AD NUMBER
AD486534
LIMITATION CHANGES
TO: **Approved for public release; distribution is unlimited.**
FROM: **Distribution authorized to DoD only; Administrative/Operational Use; 12 NOV 1964. Other requests shall be referred to Office of Naval Research, Washington, DC.**
AUTHORITY
ONR/CNA ltr 2 Dec 1970

CNA

486534

LOGISTIC FEASIBILITY SCHEDULING MODEL;
Computer Program 38-64P

By L. Emerling, G.A. Westlund
and S.A. Denenberg

Research Contribution No. 47

Research Contribution

NAVAL WARFARE ANALYSIS GROUP

Center for Naval Analyses
The Franklin Institute
WASHINGTON 25, D. C.

12 November 1964

RESEARCH CONTRIBUTION

Naval Warfare Analysis Group

CENTER FOR NAVAL ANALYSES

LOGISTIC FEASIBILITY SCHEDULING MODEL; Computer Program 38-64P

By L. Emerling, G.A. Westlund
and S.A. Denenberg

Research Contribution No. 47

L. Emerling

G.A. Westlund

S.A. Denenberg

CONTRACT NONR 3732(00)

ABSTRACT

An IBM 7090 computer program is described which simulates vehicle assignments to a priority ordered sequence of cargo units. The schedule thereby generated is used to assess the lift capability of an arbitrary vehicle inventory or to compare one vehicle inventory with another. The model can thus be used to determine the logistic feasibility of an operations plan; and if infeasibility is demonstrated, indicate where logistic augmentation or modification is most desirable. Flow charts, a listing of the FORTRAN program, and sample inputs and outputs are included.

TABLE OF CONTENTS

I. INTRODUCTION

The Logistic Feasibility Scheduling Model (LFSM), as presently formulated can be utilized with either of two functions in mind. These can best be described as requirements-oriented or capabilities-oriented. In the former application, the feasibility test accepts the schedule of requirements. It then assesses the capacity of specified logistic resources to meet the schedule. Requirements that cannot be satisfied are noted as indicating the need for augmentation of capabilities. If capabilities-oriented, the feasibility test considers the logistic resources inputs as binding, but construes the requirements schedule to be indicative of a desired time sequencing rather than as a firm schedule.

The LFSM is designed to consider all cargo units as comprising a single theater-wide priority list. Total transportation resources, both organic to the theater of operations and extra-theater, are surveyed in the determination of the lift schedule.

II. GENERAL DESCRIPTION

The LFSM simulates the following situation: A given set of cargo units must be transported from their origin ports to their delivery ports in the shortest time by a given set of vehicles. Each cargo unit has associated with it an origin port, a delivery port, a delivery time limit (DTL), a tonnage, a priority, and a time of availability. For each day in the total time period, every cargo is considered in turn (cargoes are not considered until the time they become available) on a priority basis.* Once the highest priority cargo available is chosen, all compatible vehicles are considered to determine the earliest time that each can make the lift and be accepted at the delivery port. If the earliest of these times is less than or equal to the required delivery time limit, a movement will be scheduled; otherwise, not. Normally the vehicle with earliest delivery time will be unique and is automatically scheduled. If more than one vehicle has the same earliest delivery time, within the DTL, the one with greatest cargo capacity is scheduled.

If one vehicle does not suffice to move the whole cargo unit, the weight of the unit is reduced by the capacity of the vehicle assigned, and the above cycle is repeated. In this way it is possible for a fraction of a cargo unit to be scheduled even though delivery of the full unit is not feasible within the required delivery time limit. On the other hand, if a cargo unit or its remainder does not fill the assigned vehicle and there is excess capacity, other cargo units at the same origin port are considered in order of priority. If one of these cargo units has the same delivery port as the first unit assigned, is compatible with the assigned vehicle, is available by the time the vehicle arrives at the origin port, will fit completely within the remaining vehicle capacity, and can be delivered within its DTL, it is assigned. Such a unit may be well down the priority list as compared to the first unit assigned, but its transport does not delay any higher priority movement. Fractional units are not assigned in these circumstances,

*In this model both the day of availability and the priority of the cargoes are used to determine which cargo is chosen. There exists another model, computer program 38-64P1, in which just the priority of the cargoes are used to determine which cargo is chosen.

to prevent a single cargo from being delivered at two widely separated points in time, since the model will designate for the next lift the cargo unit of highest remaining priority whether or not it is the partially delivered cargo unit.

The LFSM thus generates a feasible schedule of delivery for priority ordered cargo units, transportation being scheduled only for that fraction of each unit which can be delivered by the stipulated delivery time limit. Cargo units (partial or complete) that cannot meet the desired delivery schedules are regarded as imposing requirements for vehicle augmentation and/or port improvements.

The LFSM may be used to assess lift feasibility in any of the three modes; airlift, sealift, or combined lift. In the latter case, provision is made by the model for predetermined assignment, by the logistics planner, of those specific cargo units to be transported strictly by air and those strictly by sea. This is done, by use of a cargo compatibility index. (See section II - Inputs to the Program.) Predetermined cargo unit assignment to any of certain specific vehicles is done in the same manner.

The outputs from the computer model include trip index quantities: i.e., for each trip, the cargo unit, cargo priority, the vehicle identification number, origin port, departure time, delivery port (destination), arrival time, and tonnage for each cargo carried. Other outputs include the tonnage of each cargo unit remaining undelivered along with its DTL; a matrix of cumulative unused vehicle capacities for each type of vehicle at each delivery port; total tonnage delivered for each day in the total time period; and final delivery times for each cargo.

III. INPUTS TO THE PROGRAM

Address	Symbol	Description
1-700	CP_i	Tonnage of i^{th} cargo unit ($1 \le i \le 700$)
701-1400	PC_i	Priority of i^{th} cargo (PC_i = 1 for highest, PC_i = 2 for next lower)*
1401-2100	YP_i	Day** of availability of i^{th} cargo ($0 < YP_i \le NDAYS$) Note $YP_i \ne 0$

* It is possible for two or more cargoes to have the same priority number. If this is the case, the computer program will consider the one with the lowest address as the highest priority.

** All times are entered in days and fractions of a day according to the following convention: $0 < DAY \le 1$ is considered as first day
$1 < DAY \le 2$ is considered as second day
$NDAYS - 1 < DAY \le NDAYS$ is considered as last day
i.e., if DAY = 3.7, we are seven-tenths of the way through the fourth day.

Address	Symbol	Description
2101-2800	ZP_i	Delivery time limit (DTL) for delivery of i^{th} cargo (days)
2801-3500	OP_i	Origin port of i^{th} cargo
3501-4200	DP_i	Delivery port of i^{th} cargo
4201-4900	TC_i	Type of i^{th} cargo ($1 \leq TC_i \leq 15$)
4901-4915	$NAMAX_p$	Maximum number of air slots per day at p^{th} port ($1 \leq p \leq 15$)
4916-4930	$NSMAX_p$	Maximum number of ship slots per day at p^{th} port
4931	NCARGS	Number of cargo units ($1 \leq NCARGS \leq 700$)
4932	NVEHS	Number of vehicles ($1 \leq NVEHS \leq 500$)
4933	NDAYS	Number of days in the total time period ($1 \leq NDAYS \leq 90$)
4934	ID	Identification number of this computer run (ID is an integer such that $1 \leq ID \leq 32767$)
5001-5500	V_k	ID number of k^{th} vehicle ($1 \leq k \leq 500$) (for aircraft, $V_k \leq 1000$, for ships $V_k > 1000$)
5501-6000	VT_k	Type of k^{th} vehicle ($1 \leq VT_k \leq 15$)
6001-6500	WV_k	Day of availability of k^{th} vehicle ($0 < WV_k \leq NDAYS$) Note $WV_k \neq 0$
6501-7000	BV_k	Port of availability of k^{th} vehicle ($1 \leq BV_k \leq 15$)
1, 1, 33-15, 15, 33	$IV_{v,c}$	Compatibility of v^{th} type of vehicle with c^{th} type of cargo (if compatible, $IV_{v,c} = 1$, if not, $IV_{v,c} = 0$)
1,1,34-15,15,34	$IBV_{v,p}$	Compatibility of v^{th} type of vehicle with p^{th} port (if compatible, $IBV_{v,p} = 1$, if not $IBV_{v,p} = 0$)

III. INPUTS TO THE PROGRAM (cont'd)

Address	Symbol	Description
1, 1, 35-15, 15, 35	$DELV_{v,p}$	Processing time for v^{th} type of vehicle at p^{th} port (days)
1, 1, 36-15, 15, 50	$EV_{v,p,q}$ *	Transit time for v^{th} type of vehicle from p^{th} port to q^{th} port (days)
1, 1, 51-15, 15, 65	$HV_{v,p,q}$	Capacity of v^{th} type of vehicle from p^{th} port to q^{th} port (tons)

IV. USER'S INSTRUCTIONS

See the list of input parameters on page 2. The user should submit a data submittal form (see appendix C). These forms have space for the submitter's name, the date, the program number (38-64P), the security classification if any, and the parameter addresses and values.

Input flexibility has been attained by allowing the user to vary any or all of the parameters in a computer run. In any one set of data the parameters remain fixed, of course, but there is no programmed limit to the number of data sets a user may submit in a run. The only restriction is that each data set must terminate with one blank card, and the last set in the run must terminate with two blank cards.

A further advantage enjoyed by the user is that for each data set after the first, he need submit only those parameter values in a set that are different from those in the previous set. This is accomplished by identifying each parameter by a unique "address" in the computer memory (see appendix E). Initially every address is cleared to zero so that only non-zero input parameters need be entered.

V. SAMPLE PROBLEM

For illustrative purposes, the following sample situation was fabricated: One large cargo unit must be lifted from its origin port to a delivery port by a group of four vehicles - two ships and two aircraft. The specific parameters of

* As an example of how the subscripts are related to the data addresses: If the transit time of the second type of vehicle from the fourth port to the third port is 5 days, then $EV_{2,4,3} = 5$ and address 2,4,38 has the value 5. (See section IV - User's Instructions .)

the problem are shown in the problem submittal form in appendix C. The output is presented in appendix D.

VI. KEYPUNCH INSTRUCTIONS

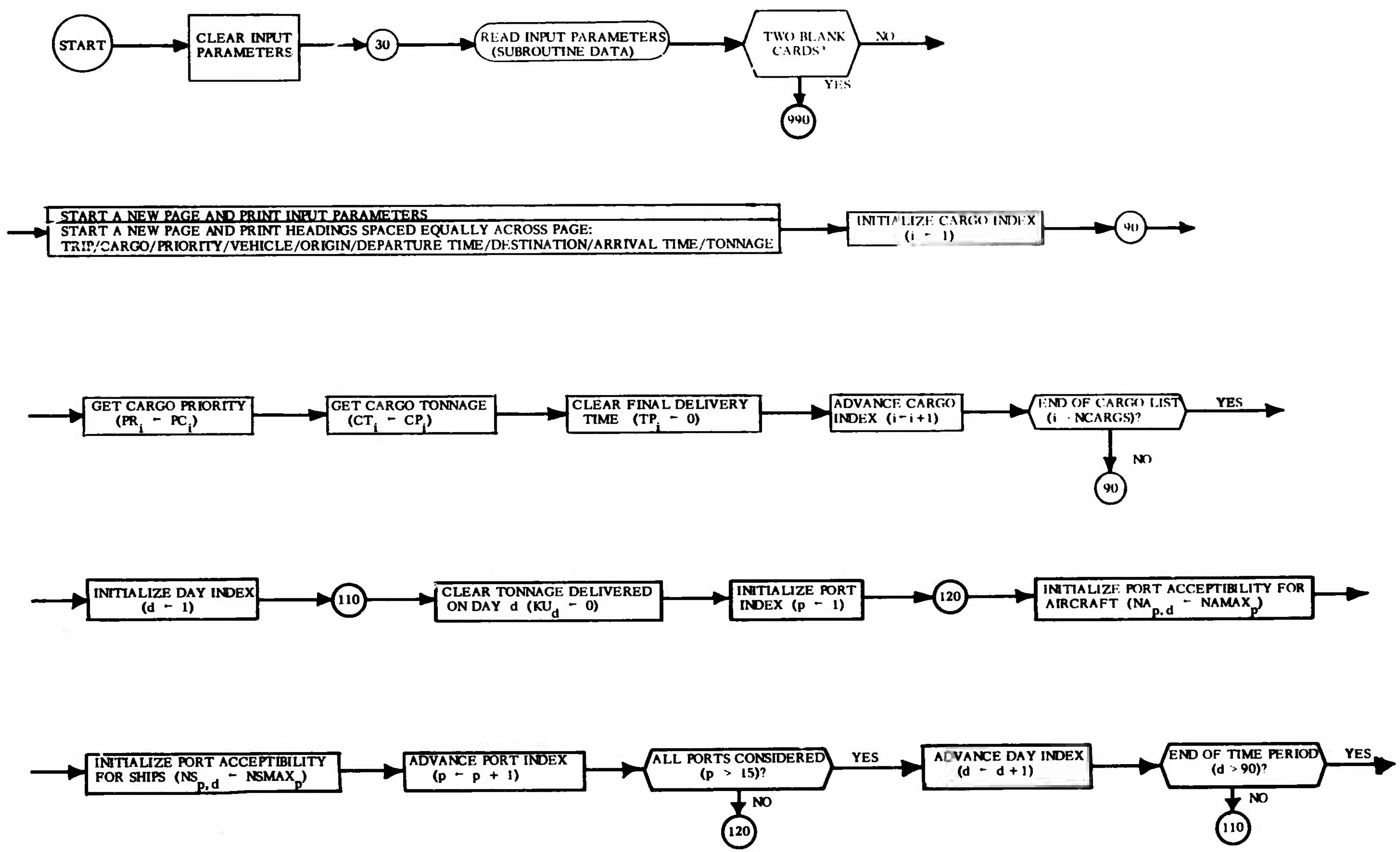
See the list of input parameters on page 2 and the sample problem submittal form in appendix C. Input card formats are described in appendix E.

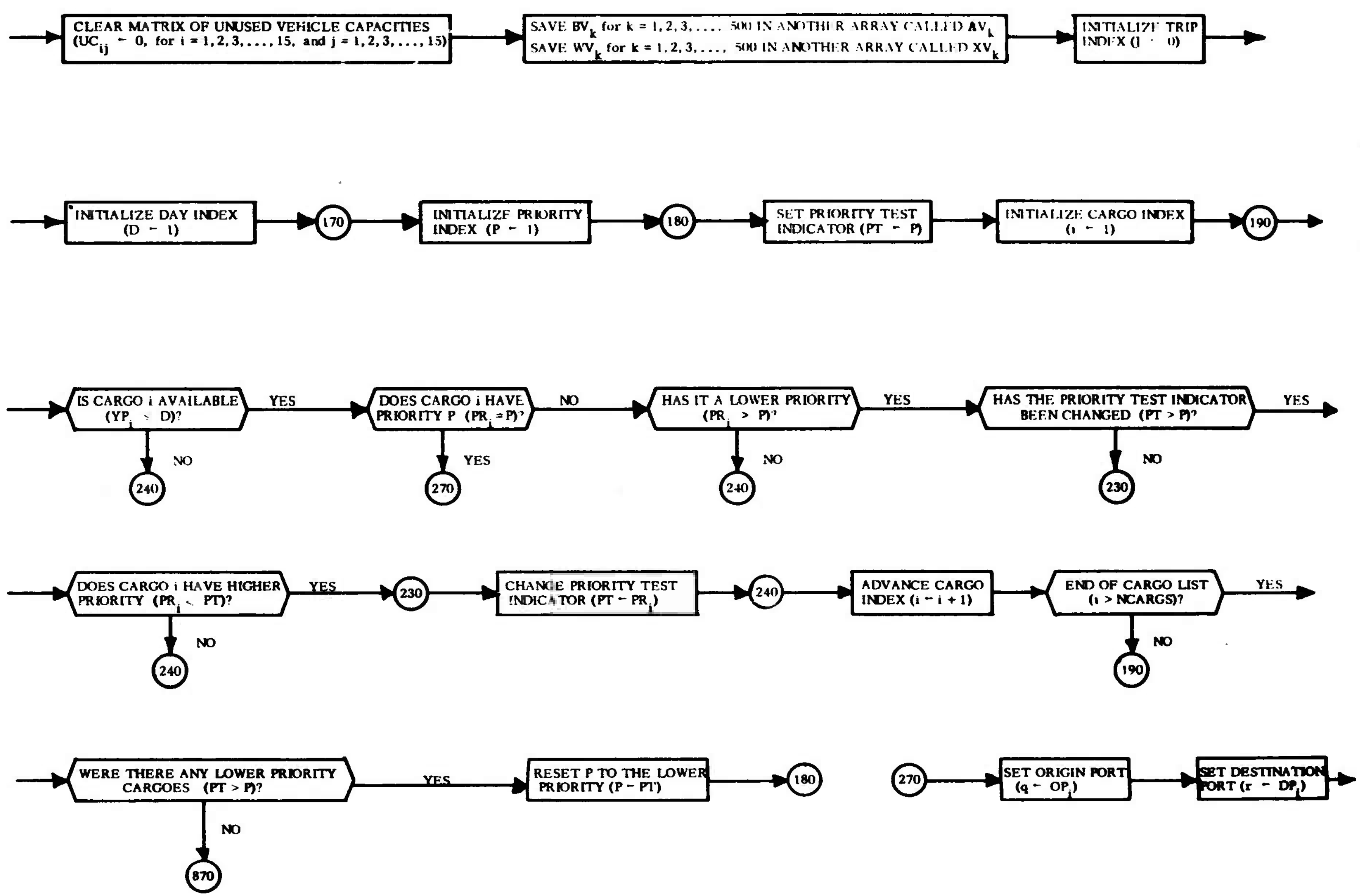
VII. OPERATOR'S INSTRUCTIONS

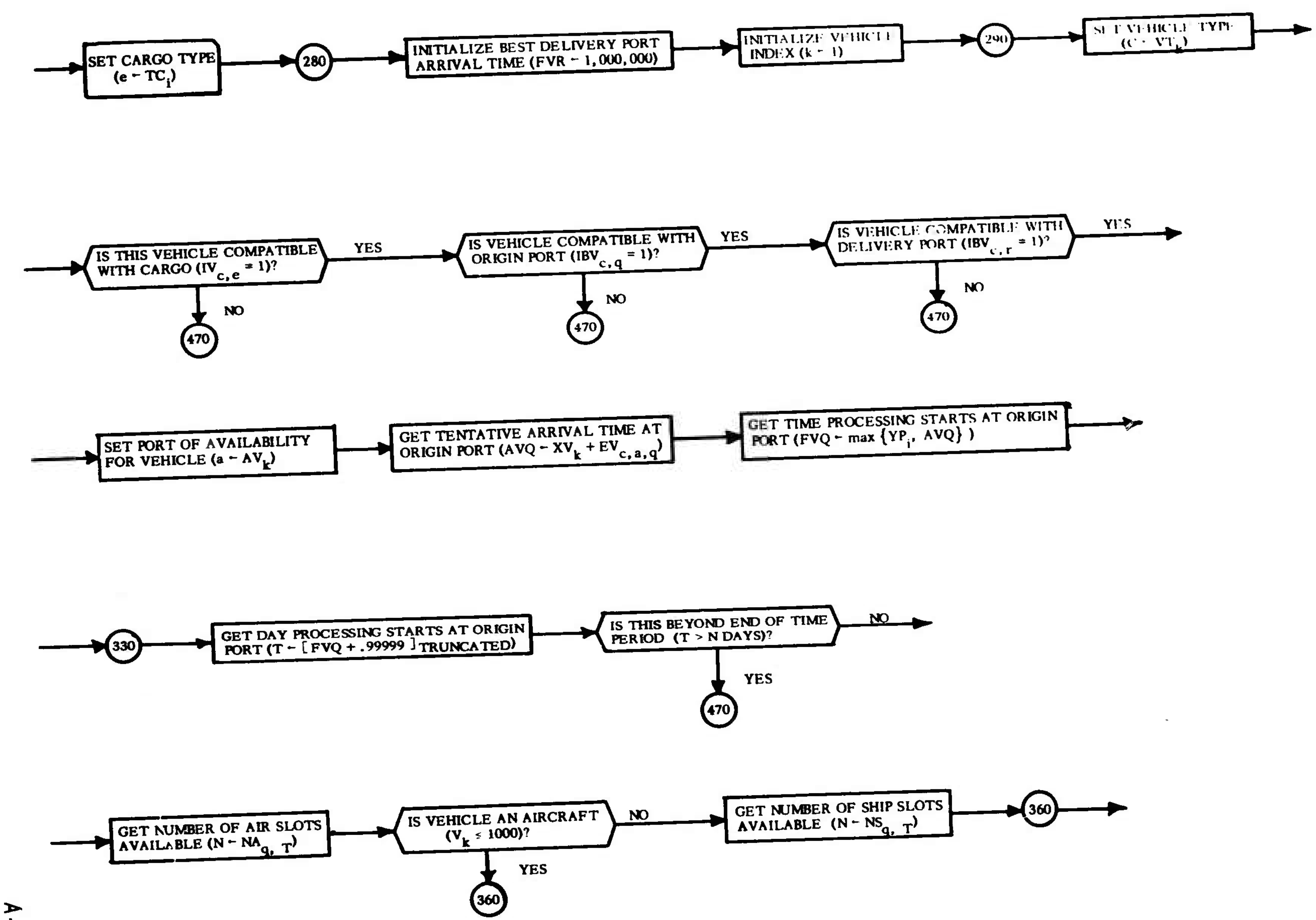
Run under control of the Bell System on the IBM 7090. No special instructions.

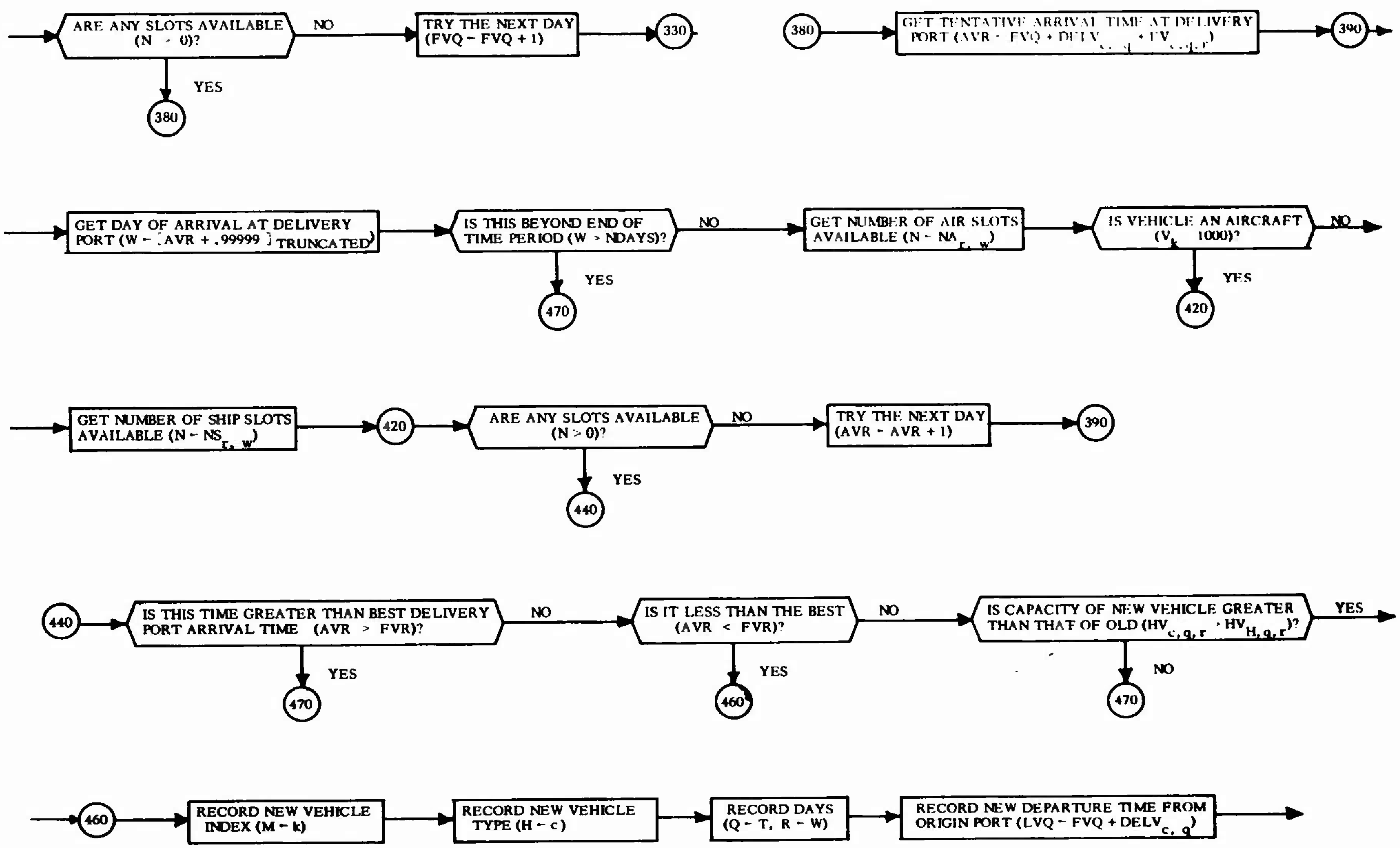
APPENDIX A

FLOW CHART

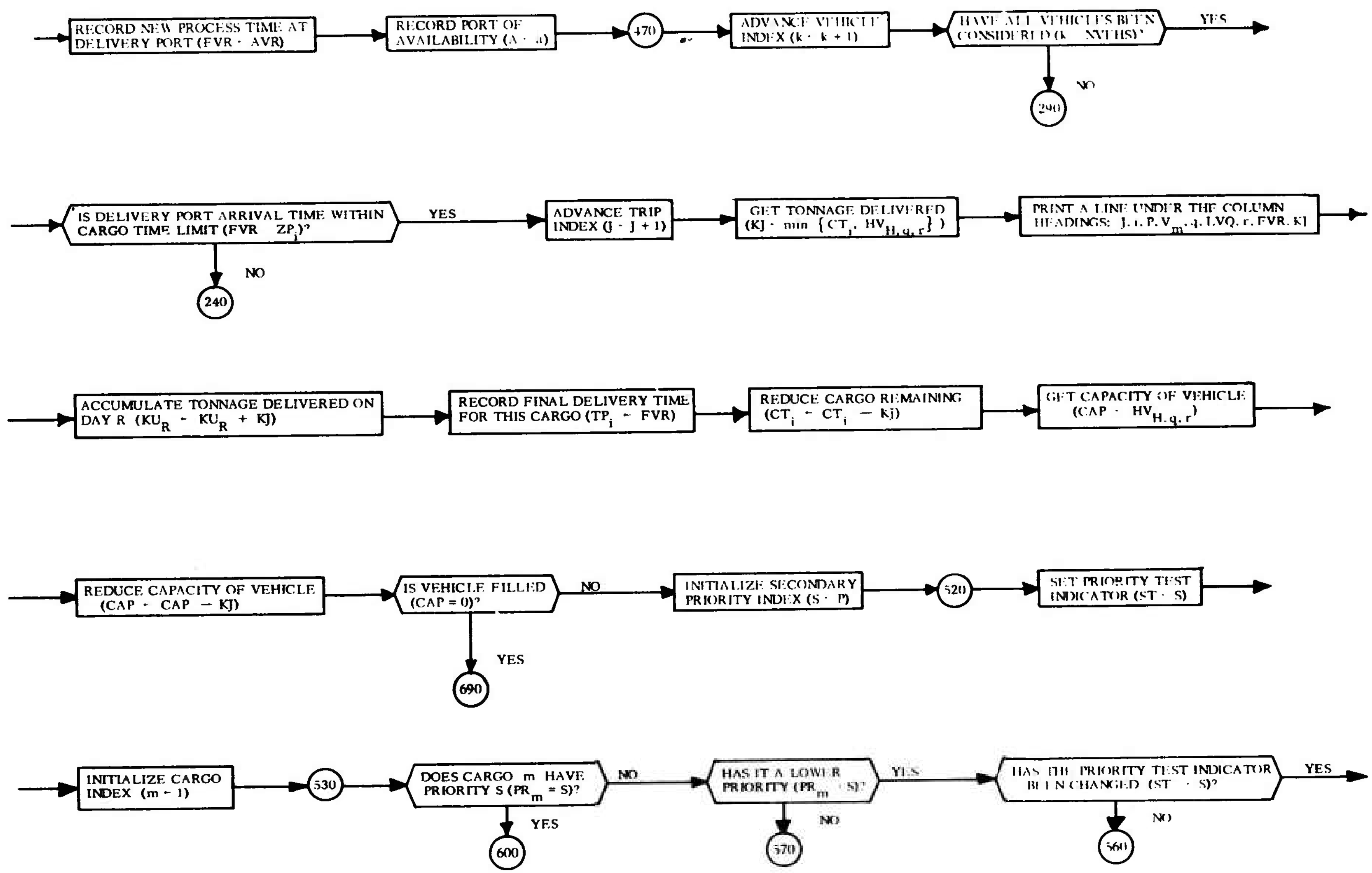

START
CLEAR INPUT PARAMETERS
30
READ INPUT PARAMETERS (SUBROUTINE DATA)
TWO BLANK CARDS?
NO
YES
990
START A NEW PAGE AND PRINT INPUT PARAMETERS
START A NEW PAGE AND PRINT HEADINGS SPACED EQUALLY ACROSS PAGE:
TRIP/CARGO/PRIORITY/VEHICLE/ORIGIN/DEPARTURE TIME/DESTINATION/ARRIVAL TIME/TONNAGE
INITIALIZE CARGO INDEX (i ← 1)
90
GET CARGO PRIORITY (PR_i ← PC_i)
GET CARGO TONNAGE (CT_i ← CP_i)
CLEAR FINAL DELIVERY TIME (TP_i ← 0)
ADVANCE CARGO INDEX (i ← i + 1)
END OF CARGO LIST (i · NCARGS)?
YES
NO
90
INITIALIZE DAY INDEX (d ← 1)
110
CLEAR TONNAGE DELIVERED ON DAY d (KU_d ← 0)
INITIALIZE PORT INDEX (p ← 1)
120
INITIALIZE PORT ACCEPTIBILITY FOR AIRCRAFT (NA_p,d ← NAMAX_p)
INITIALIZE PORT ACCEPTIBILITY FOR SHIPS (NS_p,d ← NSMAX_p)
ADVANCE PORT INDEX (p ← p + 1)
ALL PORTS CONSIDERED (p > 15)?
YES
NO
120
ADVANCE DAY INDEX (d ← d + 1)
END OF TIME PERIOD (d > 90)?
YES
NO
110

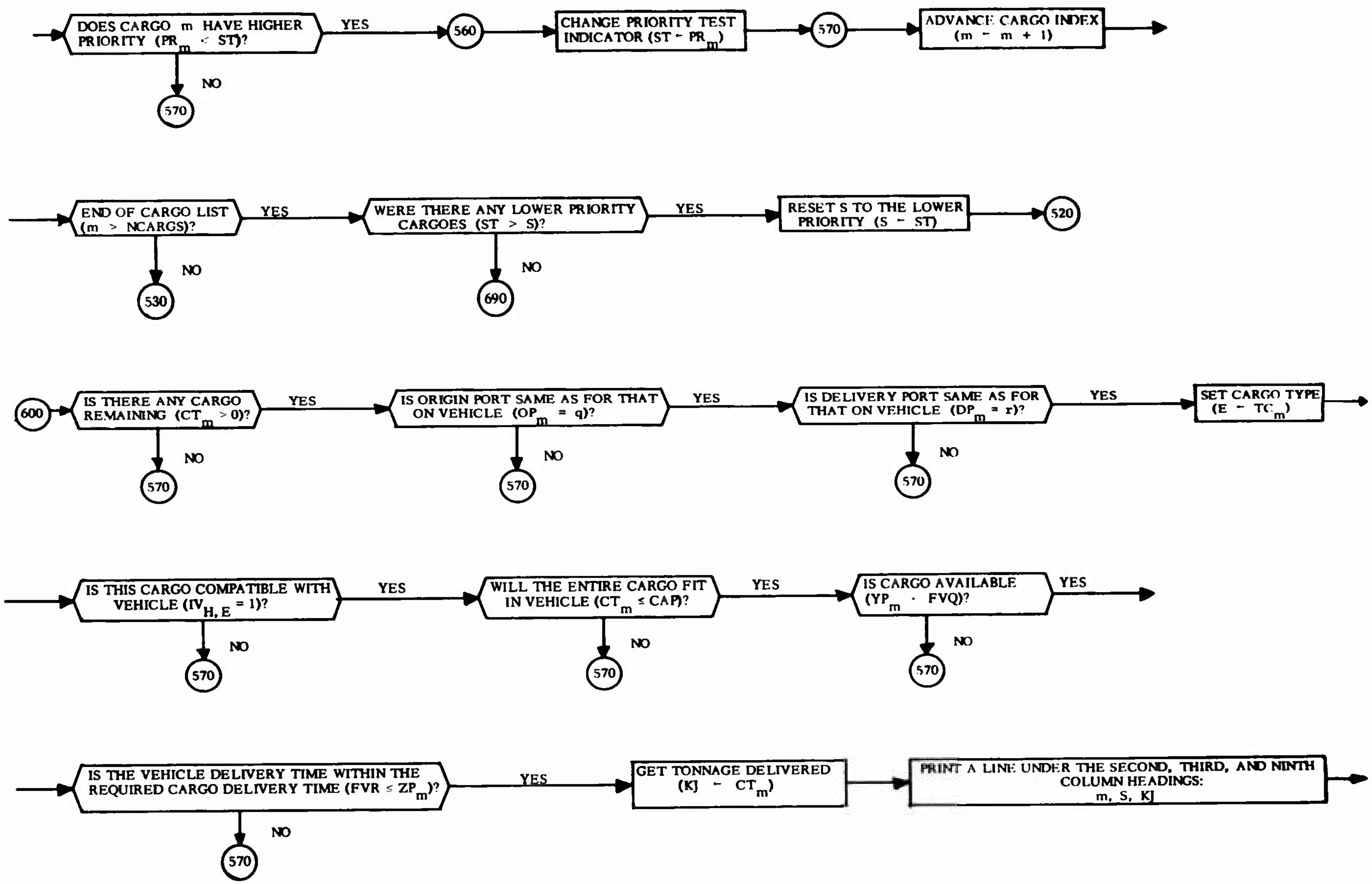

CLEAR MATRIX OF UNUSED VEHICLE CAPACITIES (UC_ij ← 0, for i = 1, 2, 3, . . . , 15, and j = 1, 2, 3, . . . , 15)
SAVE BV_k for k = 1, 2, 3, . . . , 500 IN ANOTHER ARRAY CALLED AV_k
SAVE WV_k for k = 1, 2, 3, . . . , 500 IN ANOTHER ARRAY CALLED XV_k
INITIALIZE TRIP INDEX (J ← 0)
INITIALIZE DAY INDEX (D ← 1)
170
INITIALIZE PRIORITY INDEX (P ← 1)
180
SET PRIORITY TEST INDICATOR (PT ← P)
INITIALIZE CARGO INDEX (i ← 1)
190
IS CARGO i AVAILABLE (YP_i ≤ D)?
YES
NO
240
DOES CARGO i HAVE PRIORITY P (PR_i = P)?
NO
YES
270
HAS IT A LOWER PRIORITY (PR_i > P)?
YES
NO
240
HAS THE PRIORITY TEST INDICATOR BEEN CHANGED (PT > P)?
YES
NO
230
DOES CARGO i HAVE HIGHER PRIORITY (PR_i < PT)?
YES
NO
240
230
CHANGE PRIORITY TEST INDICATOR (PT ← PR_i)
240
ADVANCE CARGO INDEX (i ← i + 1)
END OF CARGO LIST (i > NCARGS)?
YES
NO
190
WERE THERE ANY LOWER PRIORITY CARGOES (PT > P)?
YES
NO
870
RESET P TO THE LOWER PRIORITY (P ← PT)
180
270
SET ORIGIN PORT (q ← OP_i)
SET DESTINATION PORT (r ← DP_i)


SET CARGO TYPE (e ← TC_i)
280
INITIALIZE BEST DELIVERY PORT ARRIVAL TIME (FVR ← 1,000,000)
INITIALIZE VEHICLE INDEX (k ← 1)
290
SET VEHICLE TYPE (C ← VT_k)
IS THIS VEHICLE COMPATIBLE WITH CARGO (IV_c,e = 1)?
YES
NO
470
IS VEHICLE COMPATIBLE WITH ORIGIN PORT (IBV_c,q = 1)?
YES
NO
470
IS VEHICLE COMPATIBLE WITH DELIVERY PORT (IBV_c,r = 1)?
YES
NO
470
SET PORT OF AVAILABILITY FOR VEHICLE (a ← AV_k)
GET TENTATIVE ARRIVAL TIME AT ORIGIN PORT (AVQ ← XV_k + EV_c,a,q)
GET TIME PROCESSING STARTS AT ORIGIN PORT (FVQ ← max {YP_i, AVQ})
330
GET DAY PROCESSING STARTS AT ORIGIN PORT (T ← [FVQ + .99999] TRUNCATED)
IS THIS BEYOND END OF TIME PERIOD (T > N DAYS)?
NO
YES
470
GET NUMBER OF AIR SLOTS AVAILABLE (N ← NA_q, T)
IS VEHICLE AN AIRCRAFT (V_k ≤ 1000)?
NO
YES
360
GET NUMBER OF SHIP SLOTS AVAILABLE (N ← NS_q, T)
360

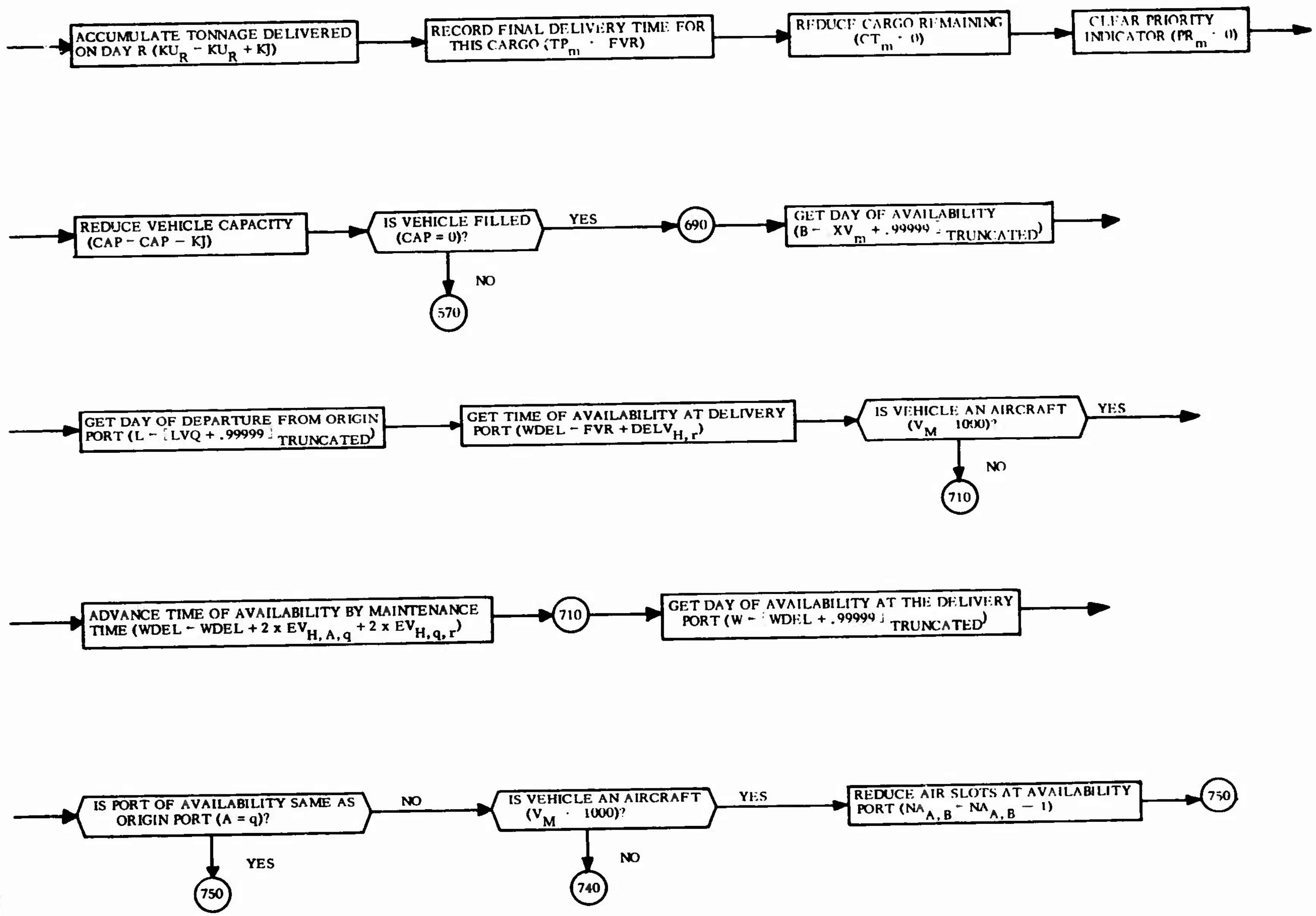


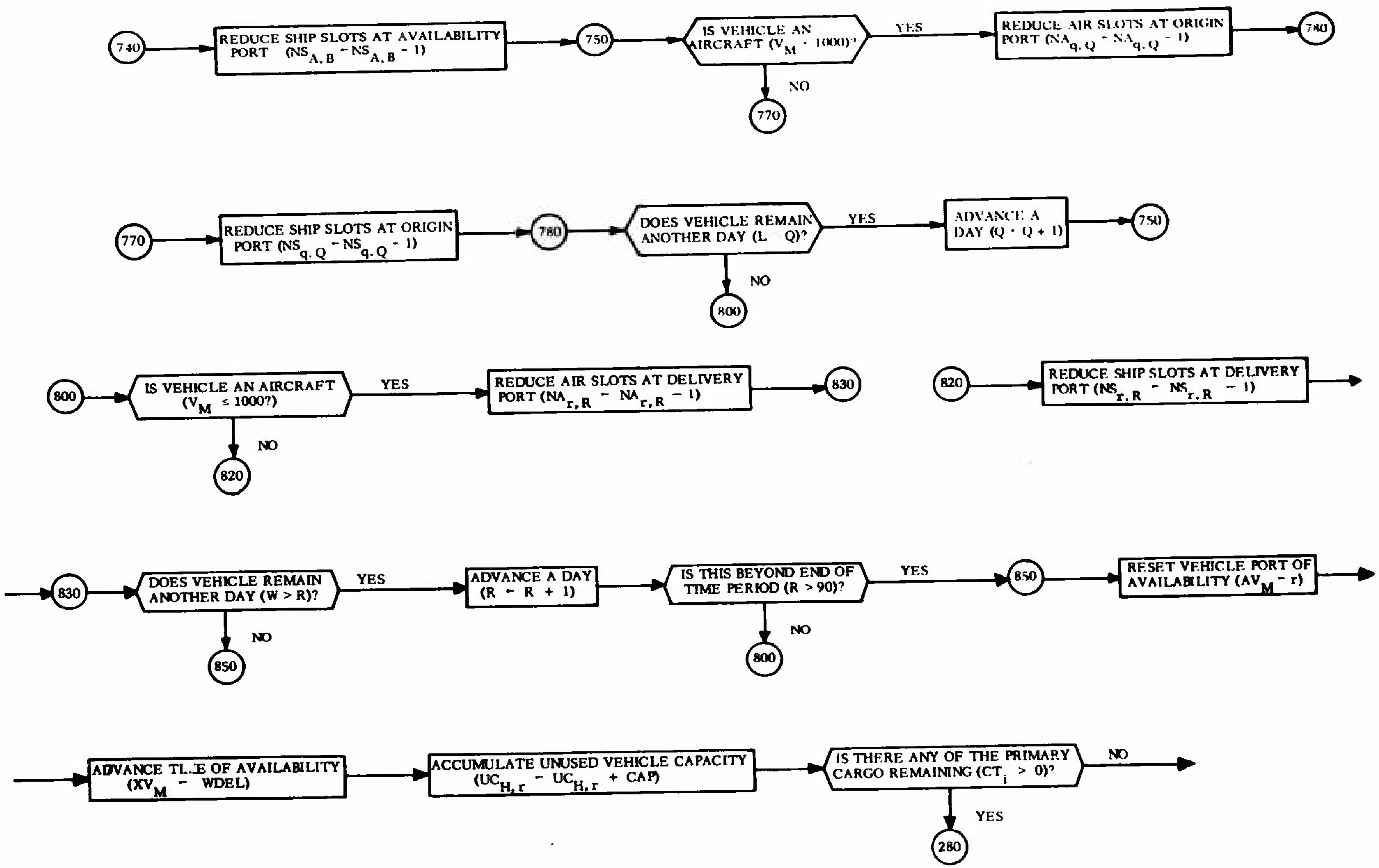
ARE ANY SLOTS AVAILABLE (N > 0)?
NO
YES
380
TRY THE NEXT DAY (FVQ ← FVQ + 1)
330
380
GET TENTATIVE ARRIVAL TIME AT DELIVERY PORT (AVR ← FVQ + DELV + FV)
390
GET DAY OF ARRIVAL AT DELIVERY PORT (W ← [AVR + .99999] TRUNCATED)
IS THIS BEYOND END OF TIME PERIOD (W > NDAYS)?
NO
YES
470
GET NUMBER OF AIR SLOTS AVAILABLE (N ← NA r, w)
IS VEHICLE AN AIRCRAFT (V k 1000)?
NO
YES
420
GET NUMBER OF SHIP SLOTS AVAILABLE (N ← NS r, w)
420
ARE ANY SLOTS AVAILABLE (N > 0)?
NO
YES
440
TRY THE NEXT DAY (AVR ← AVR + 1)
390
440
IS THIS TIME GREATER THAN BEST DELIVERY PORT ARRIVAL TIME (AVR > FVR)?
NO
YES
470
IS IT LESS THAN THE BEST (AVR < FVR)?
NO
YES
460
IS CAPACITY OF NEW VEHICLE GREATER THAN THAT OF OLD (HV c, q, r > HV H, q, r)?
YES
NO
470
460
RECORD NEW VEHICLE INDEX (M ← k)
RECORD NEW VEHICLE TYPE (H ← c)
RECORD DAYS (Q ← T, R ← W)
RECORD NEW DEPARTURE TIME FROM ORIGIN PORT (LVQ ← FVQ + DELV c, q)

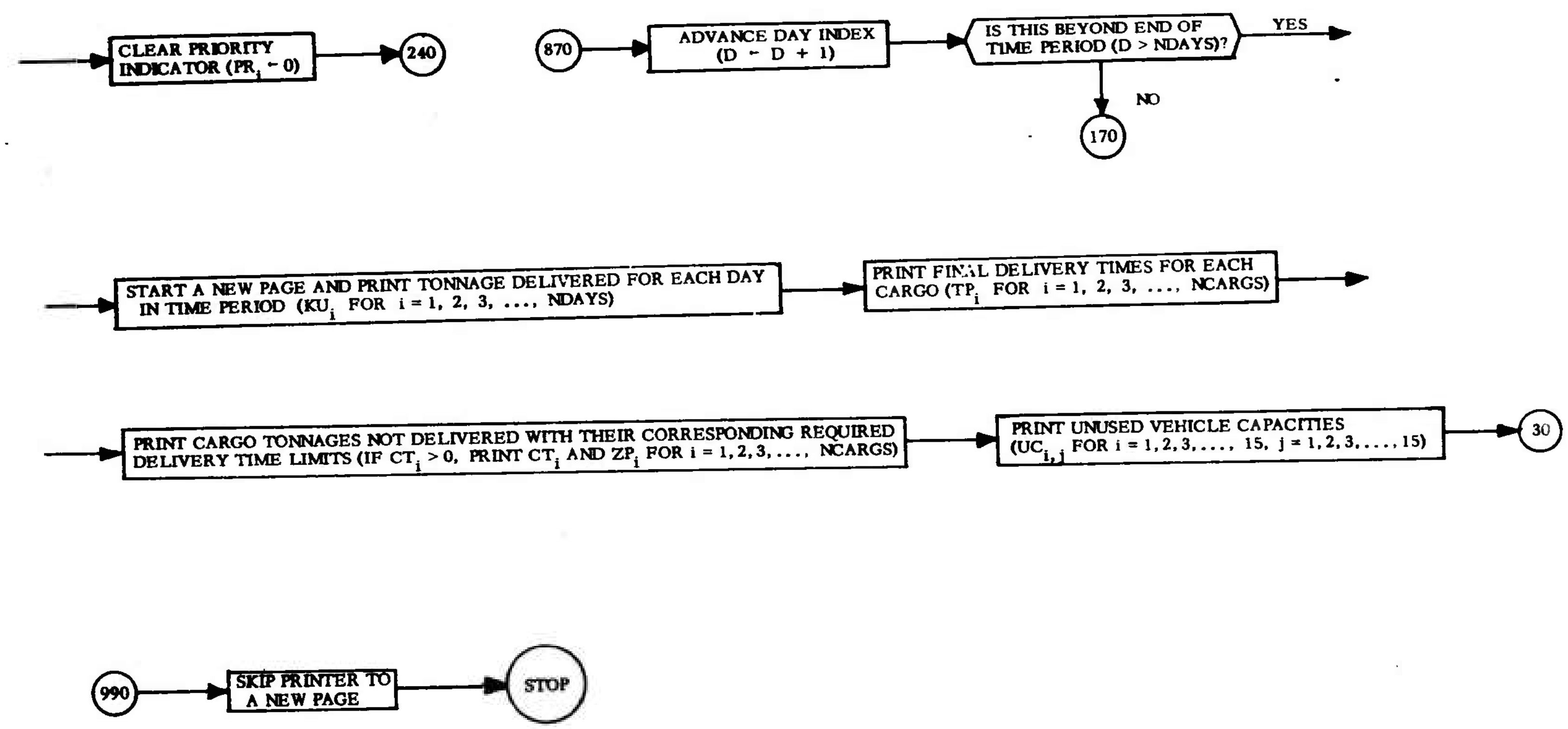

RECORD NEW PROCESS TIME AT DELIVERY PORT (FVR · AVR)
RECORD PORT OF AVAILABILITY
470
ADVANCE VEHICLE INDEX (k · k + 1)
HAVE ALL VEHICLES BEEN CONSIDERED
YES
NO
290
IS DELIVERY PORT ARRIVAL TIME WITHIN CARGO TIME LIMIT
YES
NO
240
ADVANCE TRIP INDEX (J · J + 1)
GET TONNAGE DELIVERED
PRINT A LINE UNDER THE COLUMN HEADINGS:
ACCUMULATE TONNAGE DELIVERED ON DAY R
RECORD FINAL DELIVERY TIME FOR THIS CARGO
REDUCE CARGO REMAINING
GET CAPACITY OF VEHICLE
REDUCE CAPACITY OF VEHICLE
IS VEHICLE FILLED (CAP = 0)?
NO
YES
690
INITIALIZE SECONDARY PRIORITY INDEX
520
SET PRIORITY TEST INDICATOR
INITIALIZE CARGO INDEX (m ← 1)
530
DOES CARGO m HAVE PRIORITY S
NO
YES
600
HAS IT A LOWER PRIORITY
YES
NO
570
HAS THE PRIORITY TEST INDICATOR BEEN CHANGED
YES
NO
560



DOES CARGO m HAVE HIGHER PRIORITY (PR_m < ST)?
YES
NO
570
560
CHANGE PRIORITY TEST INDICATOR (ST ← PR_m)
570
ADVANCE CARGO INDEX (m ← m + 1)
END OF CARGO LIST (m > NCARGS)?
YES
NO
530
WERE THERE ANY LOWER PRIORITY CARGOES (ST > S)?
YES
NO
690
RESET S TO THE LOWER PRIORITY (S ← ST)
520
600
IS THERE ANY CARGO REMAINING (CT_m > 0)?
YES
NO
570
IS ORIGIN PORT SAME AS FOR THAT ON VEHICLE (OP_m = q)?
YES
NO
570
IS DELIVERY PORT SAME AS FOR THAT ON VEHICLE (DP_m = r)?
YES
NO
570
SET CARGO TYPE (E ← TC_m)
IS THIS CARGO COMPATIBLE WITH VEHICLE (IV_H, E = 1)?
YES
NO
570
WILL THE ENTIRE CARGO FIT IN VEHICLE (CT_m ≤ CAP)?
YES
NO
570
IS CARGO AVAILABLE (YP_m · FVQ)?
YES
NO
570
IS THE VEHICLE DELIVERY TIME WITHIN THE REQUIRED CARGO DELIVERY TIME (FVR ≤ ZP_m)?
YES
NO
570
GET TONNAGE DELIVERED (KJ ← CT_m)
PRINT A LINE UNDER THE SECOND, THIRD, AND NINTH COLUMN HEADINGS: m, S, KJ


ACCUMULATE TONNAGE DELIVERED ON DAY R (KU_R − KU_R + KJ)
RECORD FINAL DELIVERY TIME FOR THIS CARGO (TP_m · FVR)
REDUCE CARGO REMAINING (CT_m · 0)
CLEAR PRIORITY INDICATOR (PR_m · 0)
REDUCE VEHICLE CAPACITY (CAP − CAP − KJ)
IS VEHICLE FILLED (CAP = 0)?
YES
NO
570
690
GET DAY OF AVAILABILITY (B − XV_m + .99999 TRUNCATED)
GET DAY OF DEPARTURE FROM ORIGIN PORT (L − LVQ + .99999 TRUNCATED)
GET TIME OF AVAILABILITY AT DELIVERY PORT (WDEL − FVR + $DELV_{H,r}$)
IS VEHICLE AN AIRCRAFT (V_M · 1000)?
YES
NO
710
ADVANCE TIME OF AVAILABILITY BY MAINTENANCE TIME (WDEL − WDEL + 2 x $EV_{H,A,q}$ + 2 x $EV_{H,q,r}$)
710
GET DAY OF AVAILABILITY AT THE DELIVERY PORT (W − WDEL + .99999 TRUNCATED)
IS PORT OF AVAILABILITY SAME AS ORIGIN PORT (A = q)?
YES
750
NO
IS VEHICLE AN AIRCRAFT (V_M · 1000)?
NO
740
YES
REDUCE AIR SLOTS AT AVAILABILITY PORT ($NA_{A,B}$ − $NA_{A,B}$ − 1)
750


740
REDUCE SHIP SLOTS AT AVAILABILITY PORT ($NS_{A,B} - NS_{A,B} - 1$)
750
IS VEHICLE AN AIRCRAFT (V_M · 1000)?
YES
NO
770
REDUCE AIR SLOTS AT ORIGIN PORT ($NA_{q,Q} - NA_{q,Q} - 1$)
780
770
REDUCE SHIP SLOTS AT ORIGIN PORT ($NS_{q,Q} - NS_{q,Q} - 1$)
780
DOES VEHICLE REMAIN ANOTHER DAY (L Q)?
YES
NO
800
ADVANCE A DAY (Q · Q + 1)
750
800
IS VEHICLE AN AIRCRAFT ($V_M \le 1000$?)
YES
NO
820
REDUCE AIR SLOTS AT DELIVERY PORT ($NA_{r,R} - NA_{r,R} - 1$)
830
820
REDUCE SHIP SLOTS AT DELIVERY PORT ($NS_{r,R} - NS_{r,R} - 1$)
830
DOES VEHICLE REMAIN ANOTHER DAY (W > R)?
YES
NO
850
ADVANCE A DAY (R - R + 1)
IS THIS BEYOND END OF TIME PERIOD (R > 90)?
YES
NO
800
850
RESET VEHICLE PORT OF AVAILABILITY (AV_M - r)
ADVANCE TI..E OF AVAILABILITY (XV_M - WDEL)
ACCUMULATE UNUSED VEHICLE CAPACITY ($UC_{H,r} - UC_{H,r}$ + CAP)
IS THERE ANY OF THE PRIMARY CARGO REMAINING ($CT_i > 0$)?
NO
YES
280


CLEAR PRIORITY INDICATOR (PR_i ← 0)
240
870
ADVANCE DAY INDEX (D ← D + 1)
IS THIS BEYOND END OF TIME PERIOD (D > NDAYS)?
YES
NO
170
START A NEW PAGE AND PRINT TONNAGE DELIVERED FOR EACH DAY IN TIME PERIOD (KU_i FOR i = 1, 2, 3, ..., NDAYS)
PRINT FINAL DELIVERY TIMES FOR EACH CARGO (TP_i FOR i = 1, 2, 3, ..., NCARGS)
PRINT CARGO TONNAGES NOT DELIVERED WITH THEIR CORRESPONDING REQUIRED DELIVERY TIME LIMITS (IF CT_i > 0, PRINT CT_i AND ZP_i FOR i = 1,2,3,..., NCARGS)
PRINT UNUSED VEHICLE CAPACITIES (UC_i,j FOR i = 1,2,3,..., 15, j = 1,2,3,....15)
30
990
SKIP PRINTER TO A NEW PAGE
STOP

APPENDIX B

FORTRAN STATEMENTS

```
C        LØGISTIC FEASIBILITY SCHEDULING MØDEL    38- 64P
         DIMENSIØN QQ(14630), HV(15,15,15), EV(15,15,15), DELV(15,15),
        1FIBV(15,15), FIV(15,15), BV(500), WV(500), VT(500), V(500),
        2FNSMAX(15), FNAMAX(15), TC(700), DP(700), ØP(700), ZP(700), YP(700
        3), PC(700), CP(700), AKU(90), TP(700), UC(15,15), CT(700), PR(700)
        4, FNA(15,90), FNS(15,90) ,AV(500),XV(500) ,ABC(15)
         EQUIVALENCE (QQ(1), CP(1)), (QQ(701), PC(1)), (QQ(1401), YP(1)),
        1(QQ(2101), ZP(1)),                (QQ(3501), DP(1)), (QQ(4201), TC(1))
        2, (QQ(4901), FNAMAX(1)), (QQ(4916), FNSMAX(1)), (QQ(5001), V(1)),
        3(QQ(5501), VT(1)), (QQ(6001), WV(1)),(QQ(6501), BV(1)), (QQ(7201),
        4FIV(1)), (QQ(7426), FIBV(1)), (QQ(7651), DELV(1)), (QQ(7876), EV(1
        5)), (QQ(11251), HV(1)),(QQ(2801),ØP(1))
         DØ 5 IDT=1,15
5        ABC(IDT)=(+1H )
         ABC( 3)=(+1HP)
         ABC( 6)=(+1HØ)
         ABC( 9)=(+1HR)
         ABC(12)=(+1HT)
         DØ 10 III=1,14630
10       QQ(III)=0.0
30       CALL DATA (QQ,15,15,65,IND)
         IF (IND) 40,40,990
40       IØSWCH=QQ(4935)
         IRØBØT=0
         NCARGS=QQ(4931)
         NVEHS=QQ(4932)
         NDAYS=QQ(4933)
         NRUN=QQ(4934)
         HAND=(+6H38-64P)
         HARD=(+1H1)
         IF (IØSWCH) 50,50,60
50       JCØY=0
         MAM=0
         NSØ=0
51       JCØY=JCØY+1
         MASSIV=3
         PRINT 52,HAND,NRUN,JCØY
52       FØRMAT (1H1, 39X, 18HINPUTS FØR PRØGRAM, 1X, A6,10X,4HCASE,I6,25X,
        14HPAGE, I6//19X, 1H1, 9X, 1H2, 9X, 1H3, 9X, 1H4, 9X, 1H5, 9X, 1H6,
        29X, 1H7, 9X, 1H8,9X,1H9,8X,2H10)
53       NUM=MAM
         NSØ=NUM
         MØM=MAM+1
         MAM=MØM+9
         IF (14630-MAM) 70,54,54
54       NSØ=NSØ+1
         IF (QQ(NSØ)) 57,55,57
55       IF (NSØ-MAM) 54,53,54
57       PRINT 58, NUM, (QQ(LA), LA=MØM, MAM)
58       FØRMAT(5X,I5,2X,10F10.2)
         MASSIV=MASSIV+1
         IF (MASSIV-55) 53,51,51
60       HARD=(+1H )
         PRINT 61, HAND, NRUN
61       FØRMAT (1H1, 39X,19HØUTPUTS FØR PRØGRAM, 1X, A6,10X,4HCASE,I6,25X)
70       PRINT 80 ,HARD
80       FØRMAT ( A1, 4X, 4HTRIP, 5X,5HCARGØ,5X,8HPRIØRITY,5X,7HVEHICLE,5X,
        16HØRIGIN,5X,14HDEPARTURE TIME ,5X,11HDESTINATIØN,5X,12HARRIVAL TIM
        2E,5X,7HTØNNAGE //)
```

```
      IAY=1
90    PR(IAY)=PC(IAY)
      CT(IAY)=CP(IAY)
      TP(IAY)=0.0
      IAY=IAY+1
      IF (IAY-NCARGS) 90,90,100
100   IDAY=1
110   AKU(IDAY)=0.0
      IPØRT=1
120   FNA(IPØRT, IDAY)=FNAMAX(IPØRT)
      FNS(IPØRT,IDAY)=FNSMAX(IPØRT)
      IPØRT=IPØRT+1
      IF (IPØRT-15) 120,120,130
130   IDAY=IDAY+1
      IF (IDAY-90) 110,110,140
140   DØ 150 I=1,15
      DØ 150 J=1,15
150   UC(I,J)=0.0
      DØ 160 K=1,500
      AV(K)=BV(K)
160   XV(K)=WV(K)
      JAY=0
      JDAY=1
170   PRIØRT=1.0
180   PT=PRIØRT
      IAY=1
190   IF (YP(IAY)-FLØATF(JDAY)) 200,200,240
200   IF (PR(IAY)-PRIØRT) 240,270,210
210   IF (PT-PRIØRT) 230,230,220
220   IF (PR(IAY)-PT) 230,240,240
230   PT=PR(IAY)
240   IAY=IAY+1
      IF (IAY-NCARGS) 190,190,250
250   IF (PT-PRIØRT) 870,870,260
260   PRIØRT=PT
      GØ TØ 180
270   IQ=ØP(IAY)
      IR=DP(IAY)
      IE=TC(IAY)
280   FVR=1000000.0
      KAY=1
290   IC=VT(KAY)
      IF (FIV(IC,IE)-1.0) 470,300,470
300   IF (FIBV(IC,IQ)-1.0) 470,310,470
310   IF (FIBV(IC,IR)-1.0) 470,320,470
320   IA=AV(KAY)
      AVQ=XV(KAY)+EV(IC,IA,IQ)
      FVQ=MAX1F(YP(IAY),AVQ)
330   IT=FVQ+.99999
      IF (IT-NDAYS) 340,340,470
340   NAY=FNA(IQ,IT)
      IF (V(KAY)-1000.0) 360,360,350
350   NAY=FNS(IQ,IT)
360   IF (NAY) 370,370,380
370   FVQ=FVQ+1.0
      GØ TØ 330
380   AVR=FVQ+DELV(IC,IQ)+EV(IC,IQ,IR)
390   IW=AVR+.99999
```

```
      IF (IW-NDAYS) 400,400,470
400   NAY=FNA(IR,IW)
      IF (V(KAY)-1000.0) 420,420,410
410   NAY=FNS(IR,IW)
420   IF(NAY) 430,430,440
430   AVR=AVR+1.0
      GØ TØ 390
440   IF (AVR-FVR) 460,450,470
450   IF (HV(IC,IQ,IR)-HV(IH,IQ,IR)) 470,470,460
460   IM=KAY
      IH=IC
      ILQ=IT
      ILR=IW
      FLVQ=FVQ+DELV(IC,IQ)
      FVR=AVR
      ILA=IA
470   KAY=KAY+1
      IF (KAY-NVEHS) 290,290,480
480   IF (FVR-ZP(IAY)) 490,490,240
490   JAY=JAY+1
      IFAY=FVR
      IF (FVR-FLØATF(IFAY)) 495,495,491
491   IF (FVR-(FLØATF(IFAY)+0.01)) 492,495,495
492   FFF=FLØATF(IFAY)+0.01
      GØ TØ 496
495    FFF=FVR
496   AKJ=MINIF(CT(IAY), HV(IH,IQ,IR))
      IVEH=V(IM)
       IPPPP=PRIØRT
      IF (IRØBØT-1) 499,497,497
497   PRINT 498,JAY,IVEH, FLVQ,FFF,AKJ
498   FØRMAT(  4X,I4,31X,I4,23X,F6.2,27X,F6.2,4X,F8.2)
      GØ TØ 501
499   PRINT 500, JAY,IAY,IPPPP,IVEH,IQ,FLVQ,IR,FFF,AKJ
500   FØRMAT(   4X,I4,5X,I5,8X,I5,8X,I4,7X,I4,12X,F6.2,12X,I5,10X,F6.2,4
     1X,F8.2)
501   AKU(ILR)=AKU(ILR)+AKJ
      TP(IAY)=FVR
      CT(IAY)=CT(IAY)-AKJ
      CAP=HV(IH,IQ,IR)
      CAP=CAP-AKJ
      IF (CAP) 510,690,510
510   SAY=PRIØRT
520   ST=SAY
      MC=1
530   IF (PR(MC)-SAY) 570,600,540
540   IF (ST-SAY) 560,560,550
550   IF(PR(MC)-ST) 560,570,570
560   ST=PR(MC)
570   MC=MC+1
      IF (MC-NCARGS) 530,530,580
580   IF (ST-SAY) 690,690,590
590   SAY=ST
      GØ TØ 520
600   IF (CT(MC)) 570,570,610
610   IF (ØP(MC)-FLØATF(IQ)) 570,620,570
620   IF (DP(MC)-FLØATF(IR)) 570,630,570
630   LE=TC(MC)
```

```
      IF (FIV(IH,LE)-1.0) 570,640,570
640   IF (CT(MC)-CAP) 650,650,570
650   IF (YP(MC ) -FVQ) 660,660,570
660   IF (FVR-ZP(MC)) 670,670,570
670   AKJ=CT(MC)
      ISSSS=SAY
      PRINT 680,MC,ISSSS,AKJ
680   FØRMAT (13X,I5,8X,I5,78X,F8.2)
      AKU(ILR)=AKU(ILR)+AKJ
      TP(MC)=FFF
      CT(MC)=0.0
      PR(MC)=0.0
      CAP=CAP-AKJ
      IF (CAP) 570,690,570
690   IB=XV(IM)+.99999
      LEAV=FLVQ+.99999
      WDEL=FVR+DELV(IH,IR)
      IF (V(IM)-1000.0) 700,700,710
700   WDEL=WDEL+2.0*EV(IH,ILA,IQ)+2.0*EV(IH,IQ,IR)
710   IW=WDEL+.99999
      IF (ILA-IQ) 720,750,720
720   IF (V(IM)-1000.0)  730,730,740
730   FNA(ILA,IB)=FNA(ILA,IB)-1.0
      GØ TØ 750
740   FNS(ILA,IB)=FNS(ILA,IB)-1.0
750   IF (V(IM)-1000.0) 760,760,770
760   FNA(IQ,ILQ)=FNA(IQ,ILQ)-1.0
      GØ TØ 780
770   FNS(IQ,ILQ)=FNS(IQ,ILQ)-1.0
780   IF (LEAV-ILQ) 800,800,790
790   ILQ=ILQ+1
      GØ TØ 750
800   IF (V(IM)-1000.0) 810,810,820
810   FNA(IR,ILR)=FNA(IR,ILR)-1.0
      GØ TØ 830
820   FNS(IR,ILR)=FNS(IR,ILR)-1.0
830   IF (IW-ILR) 850,850,840
840   ILR=ILR+1
      IF (ILR-90) 800,800,850
850   AV(IM)=IR
      XV(IM)=WDEL
      UC(IH,IR)=UC(IH,IR)+CAP
      IF (CT(IAY)) 860,860,851
851   IRØBØT=1
      GØ TØ 280
860    PR(IAY)=0.0
      IRØBØT=0
      GØ TØ 240
870   JDAY=JDAY+1
      IF (JDAY-NDAYS) 170,170,880
880   PRINT 890,(L,AKU(L),L=1,NDAYS)
890   FØRMAT (1H1/45X,29HTØNNAGE DELIVERED ØN EACH DAY /// 50X,3HDAY,10X
     1,7HTØNNAGE//(50X,I3,9X,F8.2))
      PRINT 900,(LC,TP(LC),LC=1,NCARGS)
900   FØRMAT (1H1/42X,35HFINAL DELIVERY TIMES FØR EACH CARGØ///50X,5HCAR
     1GØ,10X,4HTIME//(50X,I5,7X,F7.2))
      PRINT 910
910   FØRMAT(1H1/25X,56HCARGØ TØNNAGES NØT DELIVERED WITH REQUIRED DELIV
```

```
     1ERY TIME ///30X,5HCARGØ,10X,7HTØNNAGE,10X,10HTIME LIMIT//)
      II=1
930   IF (CT(II)) 960,960,940
940   PRINT 950,II,CT(II),ZP(II)
950   FØRMAT(30X,I5,9X,F9.2,12X,F7.2)
960   II=II +1
      IF (II-NCARGS) 930,930,970
970   PRINT 980
980   FØRMAT (1H1/35X,54H UNUSED VEHICLE CAPACITY BY VEHICLE TYPE FØR EA
     1CH PØRT///           58X,7HVEHICLE  //18X,1H1,6X,1H2,6X,1H3,6X,1H4,
     26X,1H5,6X,1H6,6X,1H7,6X,1H8,6X,1H9,5X,2H10,5X,2H11,5X,2H12,5X,2H13
     3,5X,2H14,5X,2H15)
      DØ 989 LB=1,15
      PRINT 981, ABC(LB),LB,(UC(LA,LB),LA=1,15)
981   FØRMAT(1X,A1,5X,I4,4X,14(F6.2,1X),F6.2)
989   CØNTINUE
      GØ TØ 30
990   PRINT 1000 ,NRUN
1000  FØRMAT (40X,18H******END ØF CASE  ,I6,6H******/1H1)
      CALL ENDJØB
      END
```

APPENDIX C

SAMPLE PROBLEM SUBMITTAL FORM

CNA COMPUTER DATA SUBMITTAL FORM

Submitted by: J. Doe Date: 1 November 1964

Program No. 38-64P Est. Time 2 Classification Uncl.

Special Instructions:

Address	Value	Address	Value	Address	Value	Address	Value
1	1000	5501	1	1,2,34	1	1,1,52	40
701	2	5502	2	2,2,34	1	2,1,52	60
1401	2	5503	3	3,2,34	1	3,1,52	5
2101	40	5504	4	4,2,34	1	4,1,52	4
2801	1	6001	3	1,1,35	0.7	1,2,51	40
3501	2	6002	3	2,1,35	1.0	2,2,51	60
4201	1	6003	1	3,1,35	0.1	3,2,51	8
4901	2	6004	1	4,1,35	0.3	4,2,51	4
4902	3	6501	1	1,2,35	0.7	— b —	
4916	3	6502	1	2,2,35	1.0	— b —	
4917	2	6503	1	3,2,35	0.1		
4931	1	6504	2	4,2,35	0.3		
4932	1	1,1,33	1	1,1,37	3.0		
4933	4	2,1,33	1	2,1,37	2.0		
4934	60	3,1,33	1	3,1,37	1.0		
5001	1001	4,1,33	1	4,1,37	0.5		
5002	1002	1,1,34	1	1,2,36	2.9		
5003	3	2,1,34	1	2,2,36	1.9		
5004	4	3,1,34	1	3,2,36	1.1		
		4,1,34	1	4,2,36	0.6		

NOTES:

1. A value of zero must be entered as 0, not left blank.
2. Decimal pts. may be omitted if understood to follow the rightmost digit.
3. The value 3×10^{-5} may be entered as .00003 or 3-5, not as 3×10^{-5}.
4. The factor portion of a value may not contain more than 8 digits.
5. The exponent portion of a value must lie within the range ±39.
6. Exponents may be omitted if zero. If not, they must be signed.
7. Blank cards should be indicated by: ———b——— .

APPENDIX D

SAMPLE PROBLEM OUTPUT

INPUTS FØR PRØGRAM 3R-64P CASE 1 PAGE

	1	2	3	4	5	6	7	8	9	10
0	1000.00	0.	0.	0.	0.	0.	0.	0.	0.	0.
700	2.00	0.	0.	0.	0.	0.	0.	0.	0.	0.
1400	2.00	0.	0.	0.	0.	0.	0.	0.	0.	0.
2100	40.00	0.	0.	0.	0.	0.	0.	0.	0.	0.
2800	1.00	0.	0.	0.	0.	0.	0.	0.	0.	0.
3500	2.00	0.	0.	0.	0.	0.	0.	0.	0.	0.
4200	1.00	0.	0.	0.	0.	0.	0.	0.	0.	0.
4900	2.00	3.00	0.	0.	0.	0.	0.	0.	0.	0.
4910	0.	0.	0.	0.	0.	3.00	2.00	0.	0.	0.
4930	1.00	4.00	60.00	1.00	0.	0.	0.	0.	0.	0.
5000	1001.00	1002.00	3.00	4.00	0.	0.	0.	0.	0.	0.
5500	1.00	2.00	3.00	4.00	0.	0.	0.	0.	0.	0.
6000	3.00	3.00	1.00	1.00	0.	0.	0.	0.	0.	0.
6500	1.00	1.00	1.00	2.00	0.	0.	0.	0.	0.	0.
7200	1.00	1.00	1.00	1.00	0.	0.	0.	0.	0.	0.
7420	0.	0.	0.	0.	0.	1.00	1.00	1.00	1.00	0.
7440	1.00	1.00	1.00	1.00	0.	0.	0.	0.	0.	0.
7650	0.70	1.00	0.10	0.30	0.	0.	0.	0.	0.	0.
7660	0.	0.	0.	0.	0.	0.70	1.00	0.10	0.30	0.
7890	2.90	1.90	1.10	0.60	0.	0.	0.	0.	0.	0.
8100	3.00	2.00	1.00	0.50	0.	0.	0.	0.	0.	0.
11260	0.	0.	0.	0.	0.	40.00	60.00	8.00	4.00	0.
11470	0.	0.	0.	0.	0.	40.00	60.00	8.00	4.00	0.

TRIP	CARGØ	PRIØRITY	VEHICLE	ØRIGIN	DEPARTURE TIME	DESTINATIØN	ARRIVAL TIME	TØNNAGE
1	1	2	4	1	2.30	2	2.80	4.00
2			3		2.10		3.10	8.00
3			1002		4.00		6.00	60.00
4			4		6.20		6.70	4.00
5			1001		3.70		6.70	40.00
6			3		6.40		7.40	8.00
7			4		10.10		10.60	4.00
8			1002		9.90		11.90	60.00
9			3		12.90		13.90	8.00
10			1001		11.00		14.00	40.00
11			4		14.00		14.50	4.00
12			1002		15.80		17.80	60.00
13			4		17.90		18.40	4.00
14			3		19.40		20.40	8.00
15			1001		18.30		21.30	40.00
16			4		21.80		22.30	4.00
17			1002		21.70		23.70	60.00
18			4		25.70		26.20	4.00
19			3		25.90		26.90	8.00
20			1001		25.60		28.60	40.00
21			1002		27.60		29.60	60.00
22			4		29.60		30.10	4.00
23			3		32.40		33.40	8.00
24			4		33.50		34.00	4.00
25			1002		33.50		35.50	60.00
26			1001		32.90		35.90	40.00
27			4		37.40		37.90	4.00
28			3		38.90		39.90	8.00

TONNAGE DELIVERED ON EACH DAY

DAY	TONNAGE
1	0.
2	0.
3	4.00
4	8.00
5	0.
6	60.00
7	44.00
8	8.00
9	0.
10	0.
11	4.00
12	60.00
13	0.
14	48.00
15	4.00
16	0.
17	0.
18	60.00
19	4.00
20	0.
21	8.00
22	40.00
23	4.00
24	60.00
25	0.
26	0.
27	12.00
28	0.
29	40.00
30	60.00
31	4.00
32	0.
33	0.
34	12.00
35	0.
36	100.00
37	0.
38	4.00
39	0.
40	8.00
41	0.
42	0.
43	0.
44	0.
45	0.
46	0.
47	0.
48	0.
49	0.
50	0.
51	0.
52	0.
53	0.
54	0.
55	0.
56	0.
57	0.
58	0.
59	0.
60	0.

FINAL DELIVERY TIMES FØR EACH CARGØ

CARGØ	TIME
1	39.90

CARGØ TØNNAGES NØT DELIVERED WITH REQUIRED DELIVERY TIME

CARGØ	TØNNAGE	TIME LIMIT
1	344.00	40.00

UNUSED VEHICLE CAPACITY BY VEHICLE TYPE FØR EACH PØRT

VEHICLE

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	1	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
	2	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
P	3	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
	4	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
	5	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
Ø	6	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
	7	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
	8	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
R	9	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
	10	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
	11	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
T	12	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
	13	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
	14	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
	15	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.

******END ØF CASE 1******

APPENDIX E

APPENDIX E

DATA SUBROUTINE

1. Introduction:

Many computer programs require the flexibility of varying any or all of the parameters in a computer run. Although FORTRAN is fairly flexible in its arithmetic and control statements, its input-output statements are quite rigid. In order to read cards for instance, considerable effort must be expended by the FORTRAN programmer in writing his input statements. This subroutine eliminates some of that tedium. The concept of a "data set" is used. A data set consists of a sequence of punched cards terminated by one blank card. A parameter deck for a computer run may consist of several data sets. Such a parameter deck is terminated by two blank cards.

2. Parameter Addresses:

The primary advantage of this subroutine over FORTRAN input statements results from the use of "parameter addresses." An address is a relative location in the computer memory. It is the subscript of an array or **matrix.** For example, in an array called X , the parameter value X_{53} would be located at address 53 . By using the parameter addresses, a user of the program need submit only those parameter values in a data set that are different from those in the previous set.

Three types of addresses are permitted by this subroutine.

(1) A numeric address consisting of one to five characters, each of which is a digit 0 - 9. Such an address (n) refers to the n^{th} element in a specified array.

(2) An alpha address consisting of one to six characters, the first of which must be alphabetic (A-Z). The remaining may be alphabetic or numeric (A-Z or 0-9). Such an address refers to the n^{th} element in a specified array ($1 \leq n \leq 26$), where the first character of the address corresponds to n as the 26 letters of the alphabet correspond to the integers 1-26.

(3) A matrix address consisting of two or more numeric fields separated by commas. For example, the address 53, 47 refers to the element in the 53^{rd} row and the 47^{th} column of a two-dimensional matrix. There is no limit to the number of dimensions in a matrix address.

3. Input Card Format:

A standard submittal form (see attachment) has been designed for the analyst. This form provides for entering parameter values with their associated addresses. The user indicates blank cards to separate data sets. The keypunch operator has the option of punching one address and value per card, or, if the addresses are sequential, of punching one address and several values on a card.

Only columns 1-72 of a card are used. Each column must contain one of the following: a digit (0-9), a "+" or " - " sign or a dash, a letter (A-Z), a period, a comma, or a blank. Each punched card must contain one parameter address. The address may start in column 1, or, if desired, may start in a later column, provided all columns before it are blank. The address is terminated by at least one blank column. Only one address is permitted on the card. Succeeding columns contain one or more parameter values, each separated by one or more blank columns. A value may be signed or unsigned. The length of the value field is variable. No blanks are permitted within a value field. A value may be punched with or without an exponent. An exponent is recognized by the presence of a plus or minus sign (or dash) between the fractional part and exponent part of the value. Decimal points (periods) may be punched in either the fractional or exponent parts of a value. If more than one value is punched on a card, those after the first will be entered at sequential addresses relative to the address of the first value.

4. Usage:

A data set is read by the use of the statement:

CALL DATA (X, I)

in a FORTRAN program for the IBM 7090. The argument X is the name of an array in the program. The argument I is an indicator set by the subroutine. This indicator may be tested by the main program upon return from the subroutine. It will have a value of 0 or 1 or 2.

0: The subroutine has read a data set. The main program will normally proceed to operate on this data.

1: The subroutine has read the second blank card which terminates the parameter deck. The main program will normally terminate at this point.

2: The subroutine has read a "bad" data card. The main program may terminate the run, or ignore the card and return to the subroutine to read the rest of the data set.

If the cards to be read contain matrix addresses, additional arguments must be included in the FORTRAN calling statement:

CALL DATA (X, D_1, D_2, D_3, ..., D_n, I)

where D_i is the i^{th} dimension of the matrix X.

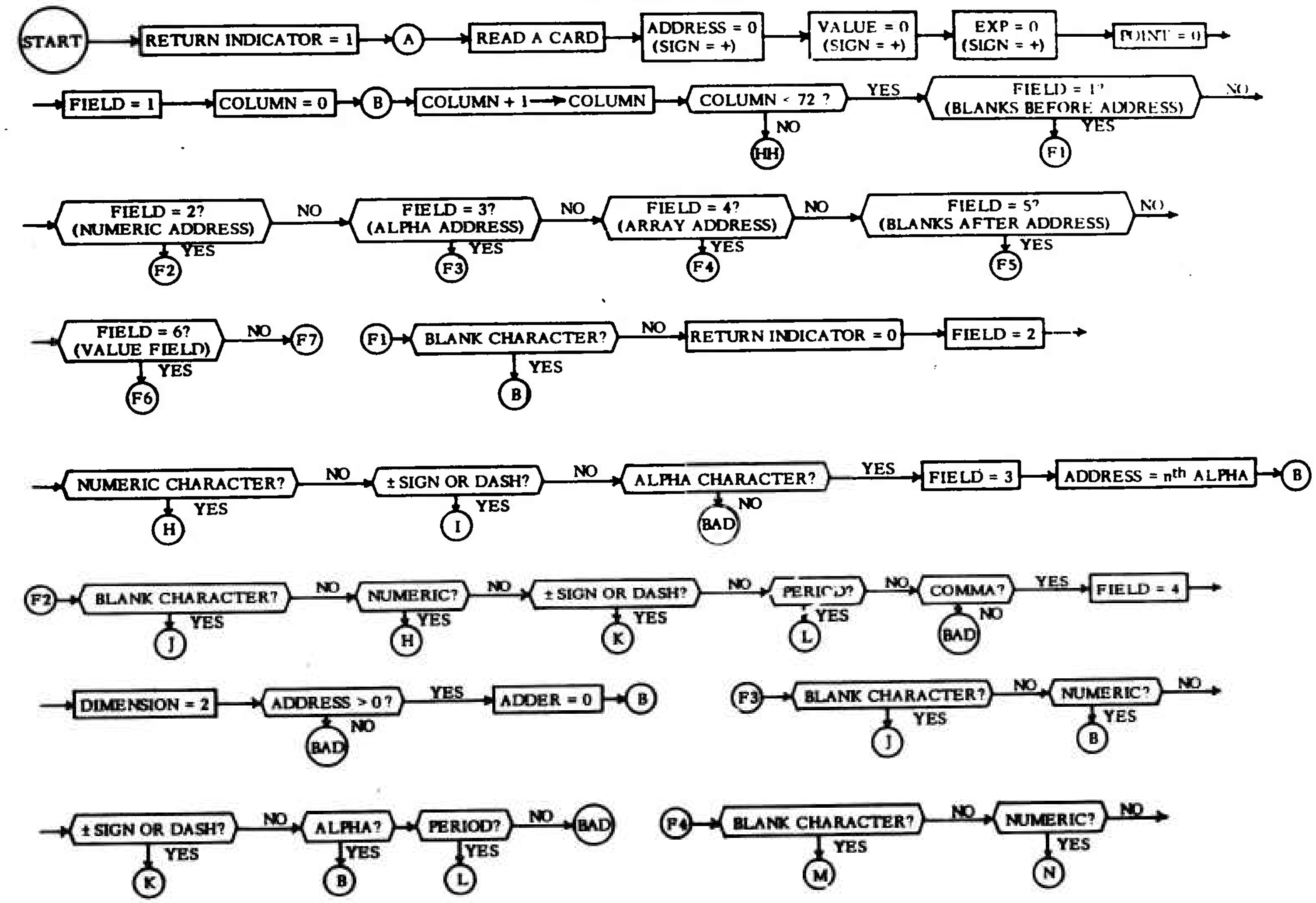
5. Method:

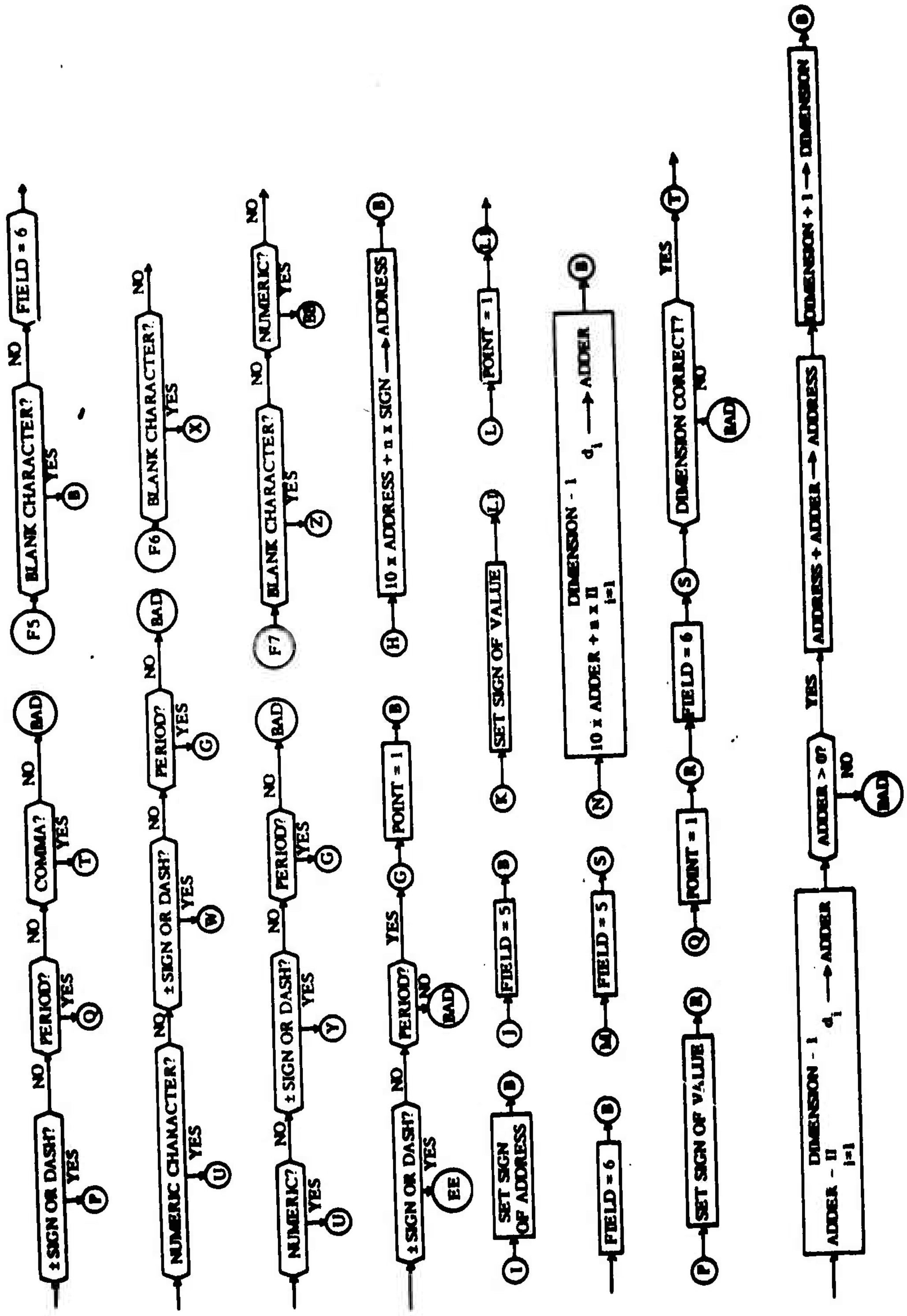
See the attached flow chart. DATA reads parameter values and loading addresses from cards. If sense switch 5 is up, it will read the values and addresses from tape (unit A2). It converts the values to floating point numbers, and stores them as elements of an array specified in the calling statement. The elements are specified by the addresses. If a card (or tape record) is read which contains non-permitted characters (see input card format above), DATA prints the statement "bad data card," followed by an image of the card itself.

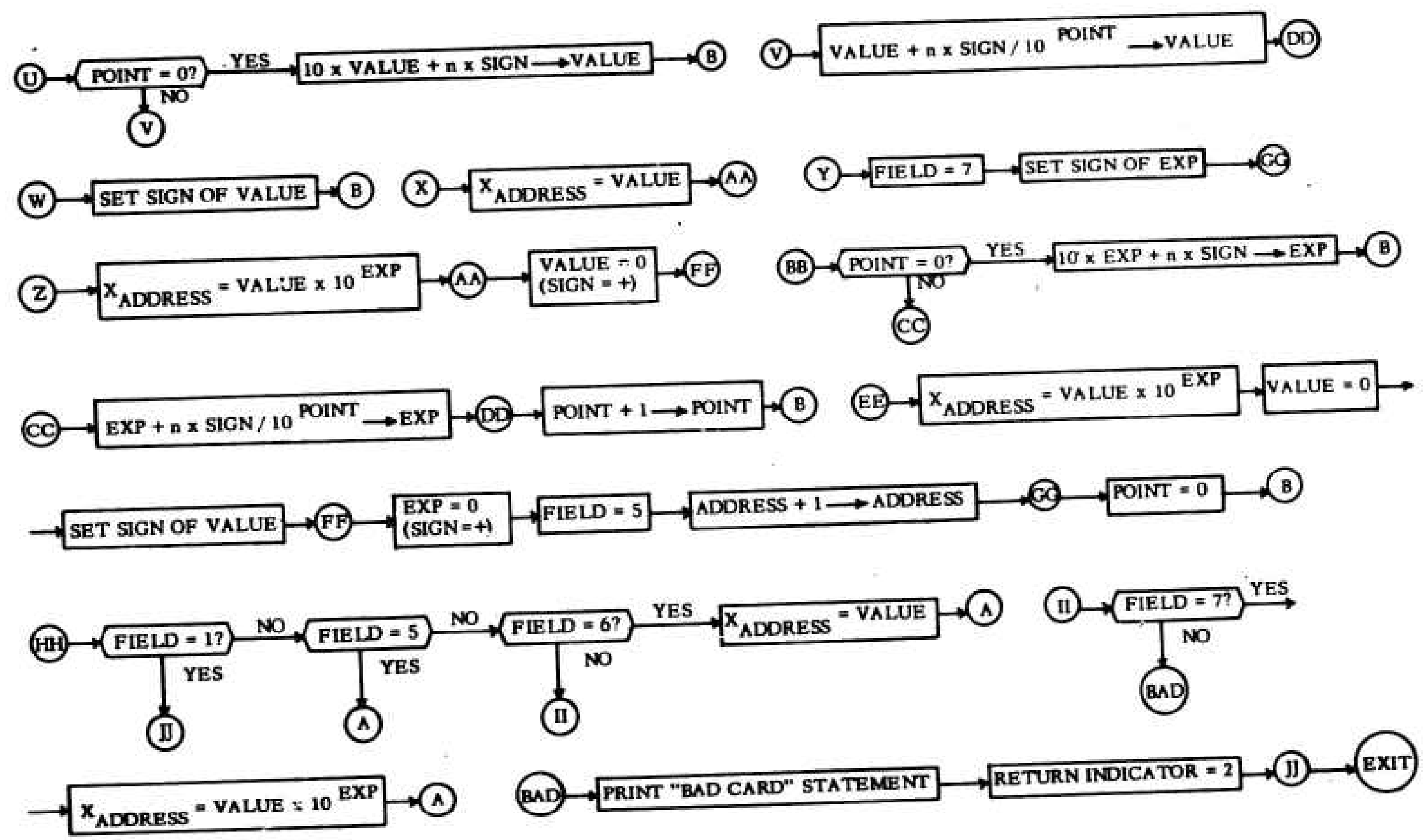
6. Coding Information:

See the symbolic listing included in this appendix. DATA is written in the 7090 FAP language. It must be used in conjunction with the BELL system. It requires 401 words storage space.

SUBROUTINE "DATA"




± SIGN OR DASH?
YES
NO
PERIOD?
COMMA?
BAD
F5
BLANK CHARACTER?
FIELD = 6
NUMERIC CHARACTER?
F6
NUMERIC?
F7
10 x ADDRESS + n x SIGN → ADDRESS
POINT = 1
SET SIGN OF ADDRESS
FIELD = 5
SET SIGN OF VALUE
DIMENSION CORRECT?
ADDER > 0?
ADDRESS + ADDER → ADDRESS
DIMENSION + 1 → DIMENSION



U
POINT = 0?
YES
NO
V
10 x VALUE + n x SIGN → VALUE
B
V
VALUE + n x SIGN / 10 POINT → VALUE
DD
W
SET SIGN OF VALUE
B
X
X ADDRESS = VALUE
AA
Y
FIELD = 7
SET SIGN OF EXP
GG
Z
X ADDRESS = VALUE x 10 EXP
AA
VALUE = 0 (SIGN = +)
FF
BB
POINT = 0?
YES
NO
CC
10 x EXP + n x SIGN → EXP
B
CC
EXP + n x SIGN / 10 POINT → EXP
DD
POINT + 1 → POINT
B
EE
X ADDRESS = VALUE x 10 EXP
VALUE = 0
SET SIGN OF VALUE
FF
EXP = 0 (SIGN = +)
FIELD = 5
ADDRESS + 1 → ADDRESS
GG
POINT = 0
B
HH
FIELD = 1?
YES
JJ
NO
FIELD = 5
YES
A
NO
FIELD = 6?
NO
II
YES
X ADDRESS = VALUE
A
II
FIELD = 7?
YES
NO
BAD
X ADDRESS = VALUE x 10 EXP
A
BAD
PRINT "BAD CARD" STATEMENT
RETURN INDICATOR = 2
JJ
EXIT

SYMBOLIC LISTING

```
        FAP
        ENTRY DATA
DATA    SXA  X1,1
        SXA  X2,2
        SXA  X4,4
        CAL  1,4
        ADD  CORE
        STO  XLOC
        AXT  1,1
        SXA  *+1,1
        CAL  **,4
        ANA  MASK
        TNZ  *+2
        TXI  *-4,1,1
        SXA  EXIT,1
        TXI  *+1,1,-1
        SXA  *+1,1
        CLA  **,4
        STA  A1
        STA  F1A
        STA  II2
        AXT  1,1               RETURN INDICATOR = 1
A1      SXD  **,1
A       TSX  HHREAD,4          READ A CARD
        PZE  CARD
        TRA  EXIT
        TRA  BAD
        STZ  ADDRES            ADDRESS = 0
        STZ  VALUE             VALUE = 0
        STZ  EXP               EXP = 0
        STZ  POINT             POINT = 0
        AXT  1,1               FIELD = 1
        SXA  FIELD,1
        AXT  13,1
A2      TNX  HH,1,1            COLUMN GT 72
        AXT  42,2
        SXA  COLUMN,2
B       LXA  COLUMN,2          COLUMN = COLUMN+1
        TNX  A2,2,6
        SXA  COLUMN,2
        LDQ  CARD+12,1
        RQL  36,2
        PXD  0,0
        LGL  6
        STO  CHARAC
        ORA  FLOAT
        FAD  FLOAT
        STO  NUMB
        AXT  42,4
        CLA  CHARAC
        CAS  TABLE+42,4
        TRA  *+2
        TRA  *+3
        TIX  *-3,4,1
```

```
        TRA  BAD
        LXA  FIELD,2
        TRA  F1+1,2
        TRA  F7          FIELD=7 (EXPONENT FIELD)
        TRA  F6          FIELD=6 (VALUE FIELD)
        TRA  F5          FIELD=5 (BLANKS AFTER ADDRESS)
        TRA  F4          FIELD=4 (ARRAY ADDRESS)
        TRA  F3          FIELD=3 (ALPHA ADDRESS)
        TRA  F2          FIELD=2 (NUMERIC ADDRESS)
F1      TXH  B,4,41      FIELD=1 (BLANKS BEFORE ADDRESS)
F1A     STZ  **          RETURN INDICATOR = 0
        AXT  2,2         FIELD = 2
        SXA  FIELD,2
        TXH  H,4,31      NUMERIC CHARACTER
        TXH  I,4,28      SIGN OR DASH
        TXL  BAD,4,2
        AXT  3,2         ALPHA CHARACTER, FIELD = 3
        SXA  FIELD,2
        TXI  *+1,4,-2    ADDRESS = NTH ALPHA
        SXA  ADDRES,4
        TRA  B
F2      TXH  J,4,41      BLANK CHARACTER
        TXH  H,4,31      NUMERIC CHARACTER
        TXH  K,4,28      SIGN OR DASH
        TXH  BAD,4,2
        TXH  L,4,1       PERIOD
        AXT  4,2         COMMA,FIELD = 4
        SXA  FIELD,2
        AXT  2,2
        SXA  DIMENS,2    DIMENSION = 2
        CLA  ADDRES      TEST ADDRESS
        TZE  BAD
        TMI  BAD
F2A     STZ  ADDER       ADDER=0
        TRA  B
F3      TXH  J,4,41      BLANK CHARACTER
        TXH  B,4,31      NUMERIC CHARACTER
        TXH  K,4,28      SIGN OR DASH
        TXH  B,4,2       ALPHA CHARACTER
        TXH  L,4,1       PERIOD
        TRA  BAD
F4      TXH  M,4,41      BLANK CHARACTER
        TXH  N,4,31      NUMERIC CHARACTER
        TXH  P,4,28      SIGN OR DASH
        TXH  BAD,4,2
        TXH  Q,4,1       PERIOD
        TRA  T           COMMA
F5      TXH  B,4,41      BLANK CHARACTER
        AXT  6,2         FIELD = 6
        SXA  FIELD,2
        TXH  U,4,31      NUMERIC CHARACTER
        TXH  W,4,28      SIGN OR DASH
        TXH  BAD,4,2
        TXH  G,4,1       PERIOD
```

```
        TRA  BAD
F6      TXH  X,4,41        BLANK CHARACTER
        TXH  U,4,31        NUMERIC CHARACTER
        TXH  Y,4,28        SIGN OR DASH
        TXH  BAD,4,2
        TXH  G,4,1         PERIOD
        TRA  BAD
F7      TXH  Z,4,41        BLANK CHARACTER
        TXH  BB,4,31       NUMERIC CHARACTER
        TXH  EE,4,28       SIGN OR DASH
        TXH  BAD,4,2
        TXL  BAD,4,1
G       AXT  1,2           PERIOD, POINT = 1
        SXA  POINT,2
        TRA  B
H       LDQ  ADDRES        ADDRESS = 10 X ADDRESS + N
        MPY  H10
        XCA
        ACL  CHARAC
        STO  ADDRES
        TRA  B
I       TXH  B,4,30        + SIGN
        CLA  ADDRES        SET SIGN OF ADDRESS
        SSM
        STO  ADDRES
        TRA  B
J       AXT  5,2           FIELD = 5
        SXA  FIELD,2
        TRA  B
K       TXH  L1,4,30       + SIGN
        CLA  VALUE         SET SIGN OF VALUE
        SSM
        STO  VALUE
        TRA  L1
L       AXT  1,2           POINT = 1
        SXA  POINT,2
L1      AXT  6,2           FIELD = 6
        SXA  FIELD,2
        TRA  B
M       AXT  5,2           FIELD = 5
        SXA  FIELD,2
        TRA  S
N       LDQ  ADDER         ADDER = 10 X ADDER + N X PROD
        MPY  H10
        STQ  ADDER
        TSX  T1,4
        MPY  CHARAC
        XCA
        ADD  ADDER
        STO  ADDER
        TRA  B
P       TXH  R,4,30        + SIGN
        CLA  VALUE         SET SIGN OF VALUE
        SSM
```

```
         STO  VALUE
         TRA  R
Q        AXT  1,2             POINT = 1
         SXA  POINT,2
R        AXT  6,2             FIELD = 6
         SXA  FIELD,2
S        LXA  EXIT,2          CHECK DIMENSION
         TXI  *+1,2,-3
         PXA  0,2
         SUB  DIMENS
         TNZ  BAD
T        TSX  T1,4            ADDER=ADDER-PROD
         CLA  ADDER
         SUB  PROD
         STO  ADDER
         TZE  BAD             CHECK ADDER
         TMI  BAD
         ADD  ADDRES
         STO  ADDRES
         CLA  DIMENS
         ADD  H1
         STO  DIMENS
         TRA  F2A
T1       SXA  T4,4            PROD = PRODUCT OF DIMENSIONS
         CLA  H1
         STO  PROD
         STA  T3.
         LXA  DIMENS,2
         TXI  *+1,2,-1
         LXA  X4,4
T2       CAL  T3
         ADD  H1
         STA  T3
T3       CLA  **,4
         STA  *+1
         LDQ  **
         RQL  18
         MPY  PROD
         STQ  PROD
         TIX  T2,2,1
T4       AXT  **,4
         TRA  1,4
U        CLA  POINT           TEST POINT
         TNZ  V
         LDQ  VALUE           VALUE = 10 X VALUE + N
         FMP  DEC10
         SSP
         FAD  NUMB
         LDQ  VALUE
         LLS  0
         STO  VALUE
         TRA  B
V        LXA  POINT,4         VALUE = VALUE + N/(10**POINT)
         CLA  NUMB
```

```
         FDP  DEC10
         XCA
         TIX  *-2,4,1
         LDQ  VALUE
         LLS  0
         FAD  VALUE
         STO  VALUE
         TRA  DD
W        TXH  B,4,30          + SIGN
         CLA  VALUE           SET SIGN OF VALUE
         SSM
         STO  VALUE
         TRA  B
X        CLA  XLOC            X(ADDRESS) = VALUE
         SUB  ADDRES
         STA  *+2
         CLA  VALUE
         STO  **
         TRA  AA
Y        AXT  7,2
         SXA  FIELD,2         FIELD = 2
         TXH  GG,4,30         + SIGN
         CLA  EXP             SET SIGN OF EXP
         SSM
         STO  EXP
         TRA  GG
Z        CLA  XLOC            X(ADDRESS) = VALUE X 10**EXP
         SUB  ADDRES
         STA  Z1
         CLA  DEC10
         LDQ  EXP
         CALL EXP(3
         XCA
         FMP  VALUE
Z1       STO  **
AA       STZ  VALUE           VALUE = 0
         TRA  FF
BB       CLA  POINT           TEST POINT
         TNZ  CC
         LDQ  EXP             EXP = 10 X EXP + N
         FMP  DEC10
         SSP
         FAD  NUMB
         LDQ  EXP
         LLS  0
         STO  EXP
         TRA  B
CC       LXA  POINT,4         EXP = EXP + N/(10**POINT)
         CLA  NUMB
         FDP  DEC10
         XCA
         TIX  *-2,4,1
         LDQ  EXP
         LLS  0
```

```
         FAD EXP
         STO EXP
DD       CLA POINT          POINT = POINT + 1
         ADD H1
         STO POINT
         TRA B
EE       CLA XLOC           X(ADDRESS) = VALUE X 10**EXP
         SUB ADDRES
         STA EE1
         CLA DEC10
         LDQ EXP
         CALL EXP(3
         XCA
         FMP VALUE
EE1      STO **
         PXD 0,0            VALUE = 0
         TXH *+2,4,30       + SIGN
         SSM                SET SIGN OF VALUE
         STO VALUE
FF       STZ EXP            EXP = 0
         AXT 5,2            FIELD = 5
         SXA FIELD,2
         CAL ADDRES         ADDRESS = ADDRESS + 1
         ADD H1
         SLW ADDRES
GG       STZ POINT          POINT = 0
         TRA B
HH       LXA FIELD,1
         TXL JJ,1,1         FIELD=1, EXIT
         TXL BAD,1,4
         TXL A,1,5          FIELD=5, READ ANOTHER CARD
         TXH II,1,6
         CLA XLOC           FIELD=6, X(ADDRESS) = VALUE
         SUB ADDRES
         STA *+2
         CLA VALUE
         STO **
         TRA A
II       TXH BAD,1,7
         CLA XLOC           FIELD=7,
         SUB ADDRES         X(ADDRESS) = VALUE X 10**EXP
         STA II1
         CLA DEC10
         LDQ EXP
         CALL EXP(3
         XCA
         FMP VALUE
II1      STO **
         TRA A
BAD      TSX HPRINT,4
         PZE PRINT,0,15
         AXT 2,1
II2      SXD **,1
X1       AXT **,1
```

```
X2     AXT **,2
X4     AXT **,4
EXIT   TRA **,4
MASK   OCT 777777700000
PRINT  BCD 3 BAD DATA CARD...
CARD   BSS 12
ADDRES HTR **
VALUE  HTR **
EXP    HTR **
POINT  HTR **
FIELD  HTR **
COLUMN HTR **
TABLE  OCT 60              BLANK
       OCT 0               0
       OCT 1               1
       OCT 2               2
       OCT 3               3
       OCT 4               4
       OCT 5               5
       OCT 6               6
       OCT 7               7
       OCT 10              8
       OCT 11              9
       OCT 20              + SIGN
       OCT 40              - SIGN
       OCT 14              DASH
       OCT 71              Z
       OCT 70              Y
       OCT 67              X
       OCT 66              W
       OCT 65              V
       OCT 64              U
       OCT 63              T
       OCT 62              S
       OCT 51              R
       OCT 50              Q
       OCT 47              P
       OCT 46              O
       OCT 45              N
       OCT 44              M
       OCT 43              L
       OCT 42              K
       OCT 41              J
       OCT 31              I
       OCT 30              H
       OCT 27              G
       OCT 26              F
       OCT 25              E
       OCT 24              D
       OCT 23              C
       OCT 22              B
       OCT 21              A
       OCT 33              PERIOD
       OCT 73              COMMA
```

```
CHARAC HTR **
DIMENS HTR **
 ADDER HTR **
 H10    HTR 10
 DEC10 DEC 10.0
 H1     HTR 1
 PROD  HTR **
 AMASK OCT 77777
 FLOAT OCT 233000000000
 NUMB  HTR **
 XLOC  HTR **
 CORE  OCT 100001
 JJ     SYN X1
        END
```

OEG COMPUTER DATA SUBMITTAL FORM

Submitted by: ______________________________ Date: ______________

Program No. ____________ Est. Time ____________ Classification ____________

Special Instructions: ____

Address	Value	Address	Value	Address	Value	Address	Value

NOTES:

1. A value of zero must be entered as 0, not left blank.
2. Decimal pts. may be omitted if understood to follow the rightmost digit.
3. The value 3×10^{-5} may be entered as .00003 or 3-5, not as 3×10^{-5}.
4. The factor portion of a value may not contain more than 8 digits.
5. The exponent portion of a value must lie within the range ±39.
6. Exponents may be omitted if zero. If not, they must be signed.
7. Blank cards should be indicated by: ———b———.

None

Security Classification

DOCUMENT CONTROL DATA - R&D	
(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)	
1. ORIGINATING ACTIVITY (Corporate author) Naval Warfare Analysis Group of the Center for Naval Analyses, The Franklin Institute	2a. REPORT SECURITY CLASSIFICATION None
	2b. GROUP
3. REPORT TITLE Logistic Feasibility Schedule Model; Computer Program 38-64P	
4. DESCRIPTIVE NOTES (Type of report and inclusive dates)	
5. AUTHOR(S) (Last name, first name, initial) Emerling, L., Westlund, G. A., Denenberg, S.A.	

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
12 November 1964	48	-----

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
NONR 3732(00)	NAVWAG Research Contribution No. 47
b. PROJECT NO. -----	
c. -----	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) ---------
d. -----	

10. AVAILABILITY/LIMITATION NOTICES

U.S. military agencies may obtain copies directly from DDC. Other qualified users shall request through the Office of the Chief of Naval Operations, Navy Department.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Office of Naval Research Washington 25, D.C.

13. ABSTRACT

An IBM 7090 computer program is described which simulates vehicle assignments to a priority ordered sequence of cargo units. The schedule thereby generated is used to assess the lift capability of an arbitrary vehicle inventory or to compare one vehicle inventory with another. The model can thus be used to determine the logistic feasibility of an operations plan; and if infeasibility is demonstrated, indicate where logistic augmentation or modification is most desirable. Flow charts, a listing of the FORTRAN program, and sample inputs and outputs are included. (U)

DD FORM 1 JAN 64 1473

None

Security Classification

None

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Computer program						
FORTRAN						
IBM 7090 (program for)						
Logistics						
Vehicle lift capability simulation						
Cargo unit priorities						
Flow charts						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through ________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through ________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through ________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, roles, and weights is optional.